Managing File Systems in Oracle® Solaris 11.2

Part No: E39021
July 2014

Copyright © 2004, 2014, Oracle and/or its affiliates. All rights reserved.

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

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

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

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

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

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

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

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

Copyright © 2004, 2014, Oracle et/ou ses affiliés. Tous droits réservés.

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

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

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

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

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

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

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

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

Contents

Using This Documentation

- **Overview** – Provides an overview of Oracle Solaris file systems which includes information about how to manage one or more file systems, and perform file system administration tasks.
- **Audience** – System administrators.
- **Required knowledge** – Basic Oracle Solaris or UNIX system administration experience and general file 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=E36784.

Access to Oracle Support

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

Feedback

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

♦ ♦ ♦ CHAPTER 1

About Managing File Systems

Managing file systems is one of your most important system administration tasks.

This is a list of the overview information in this chapter:

- "Overview of File Systems" on page 9
- "Default Oracle Solaris File Systems" on page 15
- "Overview of Mounting and Unmounting File Systems" on page 16
- "Determining a File System's Type" on page 21
- "File System Monitoring Tool (`fsstat`)" on page 21
- "Where to Find File System Management Tasks" on page 22

Overview of File Systems

A file system is a structure of directories that is used to organize and store files.

The term *file system* is used to describe the following:

- A particular type of file system: disk-based, network-based, or virtual
- The entire file tree, beginning with the root (`/`) directory
- The data structure of a disk slice or other media storage device
- A portion of a file tree structure that is attached to a mount point on the main file tree so that the files are accessible

Usually, you know from the context which meaning is intended.

The Oracle Solaris OS uses the *virtual file system* (VFS) architecture, which provides a standard interface for different file system types. The VFS architecture enables the kernel to handle basic operations, such as reading, writing, and listing files. The VFS architecture also makes it easier to add new file systems.

Types of Oracle Solaris File Systems

The Oracle Solaris OS supports three types of file systems:

- Disk-based
- Network-based
- Virtual

To identify the file system type, see "Determining a File System's Type" on page 21.

Oracle Solaris Disk-Based File Systems

Disk-based file systems are stored on physical media such as hard disks and DVDs. Disk-based file systems can be written in different formats. The available formats are described in the following table.

Disk-Based File System	Format Description
ZFS	ZFS is the default disk-based and root file system.
UFS	Legacy UNIX file system (based on the BSD Fat Fast File system that was provided in the 4.3 Tahoe release).
PxFS	Oracle Solaris Cluster software provides a cluster file system based on the Oracle Solaris Cluster Proxy File System (PxFS).
SAM-QFS	SAM-QFS is an integrated hierarchical storage manager (HSM) and storage area network (SAN) file system. SAM is the HSM storage and archive management component. QFS is the SAN scalable high performance file system component. SAM-QFS also has integrated disk volume management and tape volume management. QFS also has a write once, read many times (WORM) file system capability. QFS can be used independently of SAM when just a file system is needed. SAM requires QFS and cannot be used independently of QFS.
HSFS	High Sierra, Rock Ridge, and ISO 9660 file system. High Sierra is the first CD-ROM file system. ISO 9660 is the official standard version of the High Sierra file system. The HSFS file system is used on CD-ROMs, and is a read-only file system. Oracle Solaris HSFS supports Rock Ridge extensions to ISO 9660. When present on a CD-ROM, these extensions provide all file system features and file types, except for writability and hard links.
PCFS	PC file system, which allows read- and write- access to data and programs on DOS-formatted disks that are written for DOS-based personal computers.
UDFS	The Universal Disk Format (UDFS) file system, the industry-standard format for storing information on the optical media technology called DVD (Digital Versatile Disc or Digital Video Disc).

Each type of disk-based file system is customarily associated with a particular media device, as follows:

- ZFS or UFS with hard disk
- HSFS with CD-ROM
- PCFS with USB diskette

- UDF with DVD
- SAM-QFS with a hard disk or tape
- PxFS with a hard disk - For a cluster file system to be highly available, the underlying disk storage must be connected to more than one Oracle Solaris host. Therefore, a local file system (a file system that is stored on a host's local disk) that is made into a cluster file system is not highly available. For more information, see "Cluster File Systems" in "Oracle Solaris Cluster Concepts Guide ".

However, these associations are not restrictive. For example, DVDs can have ZFS or UFS file systems created on them.

The Universal Disk Format (UDFS) File System

For information about creating a UDFS file system on removable media, see "How to Create a File System on Removable Media" in "Managing Devices in Oracle Solaris 11.2 ".

The UDF file system is the industry-standard format for storing information on *DVD* (Digital Versatile Disc or Digital Video Disc) optical media.

The UDF file system is provided as dynamically loadable 32-bit and 64-bit modules, with system administration utilities for creating, mounting, and checking the file system on both SPARC and x86 platforms. The Oracle Solaris UDF file system works with supported ATAPI and SCSI DVD drives, CD-ROM devices, and disk drives. In addition, the Oracle Solaris UDF file system is fully compliant with the UDF 1.50 specification.

The UDF file system provides the following features:

- Ability to access the industry-standard CD-ROM and DVD-ROM media when they contain a UDF file system
- Flexibility in exchanging information across platforms and operating systems
- A mechanism for implementing new applications rich in broadcast-quality video, high-quality sound, and interactivity using the DVD video specification based on UDF format

The following features are not included in the UDF file system:

- Support for write-once media, (CD-RW), with either the sequential disk-at-once recording and incremental recording

The UDF file system requires the following:

- Supported SPARC or x86 platform
- Supported CD-ROM or DVD-ROM device

The Oracle Solaris UDF file system implementation provides the following:

- Support for industry-standard read/write UDF version 1.50
- Fully internationalized file system utilities

Network-Based File Systems

Network-based file systems can be accessed from the network. Typically, network-based file systems reside on one system, typically a server, and are accessed by other systems across the network.

With the NFS service, you can provide distributed *resources* (files or directories) by sharing them from a server and mounting them on individual clients. For more information, see "The NFS Environment" on page 18.

With the Oracle SMB service, you can provide distributed *resources* (files or directories) to Windows and Mac OS systems by sharing them from a server and mounting them on individual clients. For more information, see "The Oracle Solaris SMB Service" on page 20.

Virtual File Systems

Virtual file systems are memory-based file systems that provide access to special kernel information and facilities. Most virtual file systems do not use file system disk space. Also, some virtual file systems, such as the temporary file system (TMPFS), use the swap space on a disk.

Temporary File System

The temporary file system (TMPFS) uses local memory for file system reads and writes. Using TMPFS can improve system performance by saving the cost of reading and writing temporary files to a local disk or across the network. For example, temporary files are created when you compile a program. The OS generates a much disk activity or network activity while manipulating these files. Using TMPFS to hold these temporary files can significantly speed up their creation, manipulation, and deletion.

Files in TMPFS file systems are not permanent. These files are deleted when the file system is unmounted and when the system is shut down or rebooted.

TMPFS is the default file system type for the `/tmp` directory in the Oracle Solaris OS. You can copy or move files into or out of the `/tmp` directory, just as you would in a ZFS or UFS file system. The TMPFS file system uses swap space as a temporary backing store.

If a system with a TMPFS file system does not have adequate swap space, two problems can occur:

- The TMPFS file system can run out of space, just as regular file systems do.
- Because TMPFS allocates swap space to save file data (if necessary), some programs might not execute because of insufficient swap space.

For information about creating TMPFS file systems, see Chapter 2, "Creating and Mounting File Systems". For information about increasing swap space, see Chapter 3, "Configuring Additional Swap Space".

The Loopback File System

The loopback file system (LOFS) lets you create a new virtual file system so that you can access files by using an alternative path name. For example, you can create a loopback mount of the root `(/)` directory on `/tmp/newroot`. This loopback mounts make the entire file system hierarchy appear as if it is duplicated under `/tmp/newroot`, including any file systems mounted from NFS servers. All files will be accessible either with a path name starting from root (`/`), or with a path name that starts from `/tmp/newroot`.

For information on how to create LOFS file systems, see Chapter 2, "Creating and Mounting File Systems".

Process File System

The process file system (PROCFS) resides in memory and contains a list of active processes, by process number, in the `/proc` directory. Information in the `/proc` directory is used by commands such as `ps`. Debuggers and other development tools can also access the address space of the processes by using file system calls.

⚠ **Caution -** Do not delete files in the `/proc` directory. The deletion of processes from the `/proc` directory does not kill them. `/proc` files do not use disk space, so there is no reason to delete files from this directory.

The `/proc` directory does not require administration.

Additional Virtual File Systems

These additional types of virtual file systems are listed for your information. They do not require administration.

Virtual File System	Description
CTFS	CTFS (the contract file system) is the interface for creating, controlling, and observing contracts. A contract enhances the relationship between a process and the system resources it depends on by providing richer error reporting and (optionally) a means of delaying the removal of a resource.

Virtual File System	Description
	The service management facility (SMF) uses process contracts (a type of contract) to track the processes which compose a service, so that a failure in a part of a multi-process service can be identified as a failure of that service.
FIFOFS (first-in first-out)	Named pipe files that give processes common access to data.
FDFS (file descriptors)	Provides explicit names for opening files by using file descriptors.
MNTFS	Provides read-only access to the table of mounted file systems for the local system.
NAMEFS	Used mostly by STREAMS for dynamic mounts of file descriptors on top of files.
OBJFS	The OBJFS (object) file system describes the state of all modules currently loaded by the kernel. This file system is used by debuggers to access information about kernel symbols without having to access the kernel directly.
SHAREFS	Provides read-only access to the table of shared file systems for the local system.
SPECFS (special)	Provides access to character special devices and block devices.
SWAPFS	Used by the kernel for swapping.

Extended File Attributes

The ZFS, UFS, NFS, and TMPFS file systems have been enhanced to include extended file attributes. Extended file attributes enable application developers to associate specific attributes to a file. For example, a developer of an application used to manage a windowing system might choose to associate a display icon with a file. Extended file attributes are logically represented as files within a hidden directory that is associated with the target file.

You can use the `runat` command to add attributes and execute shell commands in the extended attribute namespace. This namespace is a hidden attribute directory that is associated with the specified file.

To use the `runat` command to add attributes to a file, you first have to create the attributes file.

```
$ runat filea cp /tmp/attrdata attr.1
```

Then, use the `runat` command to list the attributes of the file.

```
$ runat filea ls -l
```

For more information, see `runat`(1).

Many Oracle Solaris file system commands have been modified to support file system attributes by providing an attribute-aware option. Use this option to query, copy, or find file attributes. For more information, see the specific man page for each file system command.

Swap Space

The Oracle Solaris OS uses some disk slices for temporary storage rather than for file systems. These slices are called *swap* slices, or *swap space*. Swap space is used for virtual memory storage areas when the system does not have enough physical memory to handle current processes.

Since many applications rely on swap space, you should know how to plan for, monitor, and add more swap space, when needed. For an overview about swap space and instructions for adding swap space, see Chapter 3, "Configuring Additional Swap Space".

Default Oracle Solaris File Systems

Oracle Solaris ZFS, a revolutionary new file system, provides simple administration, transactional semantics, end-to-end data integrity, and immense scalability.

The ZFS file system is hierarchical, starting with the root directory (/) and continuing downwards through a number of directories. The Oracle Solaris installation process enables you to install a default set of directories and uses a set of conventions to group similar types of files together.

In addition, ZFS provides the following administration features:

- Device management support
- Persistent snapshots and cloning features
- Quotas that can be set for file systems
- ACL-based access control
- Storage pool space reservations for file systems
- Support for Oracle Solaris systems that have zones installed

For more information about using ZFS, see "Managing ZFS File Systems in Oracle Solaris 11.2 ".

For a brief overview of Oracle Solaris file systems and directories, see `filesystem`(5).

The following table provides a summary of the default Oracle Solaris file systems.

TABLE 1-1 The Default Oracle Solaris File Systems

File System or Directory	File System Type	Description
root (/)	ZFS	The top of the hierarchical file tree. The root (/) directory contains the directories and files that are critical for system operation, such as the kernel, the device drivers, and the programs used to boot the system. The root (/) directory also contains the mount point

File System or Directory	File System Type	Description
		directories where local and remote file systems can be attached to the file tree.
`/usr`	ZFS	System files and directories that can be shared with other users. Files that run only on certain types of systems are in the `/usr` directory (for example, SPARC executables). Files that can be used on all types of systems, such as the man pages, might be placed in the `/usr/share` directory.
`/export/home` or `/home`	NFS or ZFS	The mount point for user home directories, which store user work files. By default, the `/home` directory is an automounted file system.
`/var`	ZFS	System files and directories that are likely to change or grow over the life of the local system. These include system logs, such as `vi` and `ex` backup files.
`/opt`	NFS or ZFS	Optional mount point for third-party software. On some systems, the `/opt` directory might be a UFS file system or ZFS file system.
`/tmp`	TMPFS	Temporary files, which are removed each time the system is booted or the `/tmp` file system is unmounted.
`/proc`	PROCFS	A list of active processes, by process number.
`/etc/mnttab`	MNTFS	A virtual file system that provides read-only access to the table of mounted file systems for the local system.
`/system/volatile`	TMPFS	A memory-based file system for storing temporary files that are not needed after the system is booted.
`/system/contract`	CTFS	A virtual file system that maintains contract information.
`/system/object`	OBJFS	A virtual file system that is used by debuggers to access information about kernel symbols without having to access the kernel directly.

Overview of Mounting and Unmounting File Systems

Before you can access the files on a file system, you need to mount the file system. When you mount a file system, you attach that file system to a directory (*mount point*) and make it available to the system. The root (`/`) file system is always mounted. Any other file system can be connected or disconnected from the root (`/`) file system.

Most file systems are automatically mounted by SMF services at system boot time. Generally, you do not need to mount or unmount file systems manually. For more information about mounting different file system types, see “Mounting and Unmounting Oracle Solaris File Systems” on page 27.

When you mount a file system, any files or directories in the underlying mount point directory are unavailable as long as the file system is mounted. These files are not permanently affected by the mounting process. They become available again when the file system is unmounted. However, mount directories are typically empty because you usually do not want to obscure existing files.

For step-by-step instructions on how to mount file systems, see "Mounting and Unmounting Oracle Solaris File Systems" on page 27.

The Mounted File System Table

Whenever you mount or unmount a file system, the `/etc/mnttab` (mount table) file is modified with the list of currently mounted file systems. You can display the contents of this file by using the `cat` or `more` commands. However, you cannot edit this file. Here is an example of an `/etc/mnttab` file:

```
$ more /etc/mnttab
rpool/ROOT/zfsBE    /       zfs     dev=3390002     0
/devices        /devices        devfs   dev=8580000     1337114941
/dev    /dev    dev     dev=85c0000     1337114941
ctfs    /system/contract        ctfs    dev=8680001     1337114941
proc    /proc   proc    dev=8600000     1337114941
mnttab  /etc/mnttab     mntfs   dev=86c0001     1337114941
swap    /system/volatile        tmpfs   xattr,dev=8700001   1337114941
objfs   /system/object  objfs   dev=8740001     1337114941
sharefs /etc/dfs/sharetab       sharefs dev=8780001     1337114941
/usr/lib/libc/libc_hwcap2.so.1  /lib/libc.so.1  lofs dev=3390002  13371149
fd      /dev/fd fd      rw,dev=8880001  1337114969
rpool/ROOT/zfsBE/var        /var    zfs     rw,devices,
setuid,nonbmand,exec,
rstchown,xattr,atime,dev=3390003        1337114969
swap    /tmp    tmpfs   xattr,dev=8700002       1337114969
rpool/VARSHARE  /var/share      zfs     rw,devices,setuid,nonbmand,exec,
rstchown,xattr,atime,dev=3390004        1337114969
```

The Virtual File System Table

Most file systems are mounted automatically by an SMF service at system boot time.

You might need to edit the `/etc/vfstab` file to mount legacy or remote file systems or to make changes to the ZFS swap volume. For information about changing a ZFS swap volume, see Chapter 3, "Configuring Additional Swap Space".

To add an entry for mounting a legacy or remote file system, the information you need to specify is as follows:

- The device or the NFS server where the file system resides
- The file system mount point
- File system type

- Whether you want the file system to mount automatically when the system boots (by using the `mountall` command)
- Any mount options

The following `vfstab` example is from a system that has a ZFS root file system. In addition, this system is mounting a remote file system, `/users/data`, from the NFS server, `neo`.

```
# cat /etc/vfstab
#device         device          mount           FS      fsck    mount   mount
#to mount       to fsck         point           type    pass    at boot options
#
fd              -               /dev/fd         fd      -       no      -
/proc           -               /proc           proc    -       no      -
/dev/zvol/dsk/rpool/swap -      -               swap    -       no      -
/devices        -              /devices         devfs   -       no      -
sharefs         -              /etc/dfs/sharetabsharefs -       no      -
ctfs            -              /system/contract ctfs    -       no      -
objfs           -              /system/object   objfs   -       no      -
swap            -              /tmp             tmpfs   -       yes     -
neo:/users/data  -          /data            nfs     -       yes     -
```

ZFS file systems are mounted automatically by the SMF service at boot time. You can mount ZFS file systems from the `vfstab` by using the legacy mount feature.

For descriptions of each `/etc/vfstab` field and information on how to edit and use the file, see "How to Add an Entry to the `/etc/vfstab` File" on page 31.

The NFS Environment

NFS is a distributed file system service that can be used to share *resources* (files or directories) from one system, typically a server, with other systems on the network. For example, you might want to share third-party applications or source files with users on other systems.

NFS makes the actual physical location of the resource irrelevant to the user. Instead of placing copies of commonly used files on every system, NFS allows you to place one copy on one system's disk and let all other systems access it from the network. Under NFS, remote files are virtually indistinguishable from local files.

For more information, see "Managing Network File Systems in Oracle Solaris 11.2 ".

A system becomes an NFS server if it has resources to share on the network. A server keeps a list of currently shared resources and their access restrictions (such as read/write or read-only access).

When you share a resource, you make it available for mounting by remote systems.

You can share a resource in these ways:

- Create a ZFS share by setting the ZFS `share.nfs` property. For example:

  ```
  # zfs set share.nfs=on tank/home
  ```
- Create a legacy share by using the `share` command.

  ```
  # share -F nfs /ufsfs
  ```

For a complete description of NFS, see "Managing Network File Systems in Oracle Solaris 11.2 ".

NFS Version 4

Oracle's implementation of the NFS version 4 distributed file access protocol is included in the Oracle Solaris release.

NFS version 4 integrates file access, file locking, and mount protocols into a single, unified protocol to ease traversal through a firewall and improve security. The Oracle Solaris implementation of NFS version 4 is fully integrated with Kerberos V5, also known as SEAM, thus providing authentication, integrity, and privacy. NFS version 4 also enables the negotiation of security flavors to be used between the client and the server. With NFS version 4, a server can offer different security flavors for different file systems.

For more information about NFS Version 4 features, see "Managing Network File Systems in Oracle Solaris 11.2 ".

Sharing File System Data Across Boot Environments

A mechanism that enables automatic data sharing across different boot environments is available from Oracle Solaris 11.1 release. These shared directories are stored in the `rpool/VARSHARE` file system that is mounted at `/var/share`. Placement of shared data in the `/var` directory reduces the amount of space needed for all boot environments.

For example:

```
# ls /var/share
audit cores crash mail
```

The `/var/share` file system generally requires no administration, with the exception of ensuring that the `/var` components do not fill the root file system.

Symbolic links are automatically created from `/var` to the `/var/share` components listed above for compatibility purposes. For more information, see `datasets`(5).

Automounting (autofs)

You can mount NFS file system resources by using a client-side service called *automounting* (or *autofs*). The autofs service enables a system to automatically mount and unmount NFS resources whenever you access them. The resource remains mounted as long as you remain in the directory and are using a file within that directory. If the resource is not accessed for a certain period of time, it is automatically unmounted.

The autofs service provides the following features:

- NFS resources don't need to be mounted when the system boots, which saves booting time.
- Users don't need to know the root password to mount and unmount NFS resources.
- Network traffic might be reduced because NFS resources are mounted only when they are in use.

The autofs service is initialized by the `automount` utility, which runs automatically when a system is booted. The `automountd` daemon runs continuously and is responsible for the mounting and unmounting of NFS file systems on an as-needed basis. By default, the `/home` file system is mounted by the `automount` daemon.

With autofs, you can specify multiple servers to provide the same file system. This way, if one of these servers is down, autofs can try to mount the file system from another machine.

For complete information on how to set up and administer autofs, see “Managing Network File Systems in Oracle Solaris 11.2 ”.

The Oracle Solaris SMB Service

The Oracle Solaris OS provides a Server Message Block (SMB) protocol server and client implementation that includes support for numerous SMB dialects including NT LM 0.12 and Common Internet File System (CIFS). The terms CIFS and SMB can be considered interchangeable.

The Solaris SMB server allows a native Oracle Solaris system to serve files as SMB *shares* to SMB enabled clients that mount the file system shares. A Windows, Mac OS, or Solaris client can interoperate with the Solaris SMB server as it would with a Windows server. A Solaris SMB server can operate in either workgroup mode or in domain mode. In workgroup mode, the Solaris SMB server is responsible for authenticating users locally when access is requested to shared resources. This authentication process is referred to as local login. In domain mode, the Solaris SMB server uses pass-through authentication, in which user authentication is delegated to a domain controller.

For more information, see *Managing SMB File Sharing and Windows Interoperability in Oracle Solaris 11.2*.

Determining a File System's Type

These commands work whether or not the file system is mounted.

If you have the raw device name of a disk slice, you can use the fstyp or the df command to determine a file system's type. For more information, see fstyp(1M) or df(1M).

EXAMPLE 1-1 How to Determine a File System's Type

The following example uses the fstyp command to determine the file system type.

```
# fstyp /dev/rdsk/c0t0d0s0
zfs
```

The following example uses the df -n command to display a system's file system types.

```
# df -n
/                   : zfs
/devices            : devfs
/dev                : dev
/system/contract    : ctfs
/proc               : proc
/etc/mnttab         : mntfs
/system/volatile    : tmpfs
/system/object      : objfs
/etc/dfs/sharetab   : sharefs
/dev/fd             : fd
/var                : zfs
/tmp                : tmpfs
/var/share          : zfs
/export             : zfs
/export/home        : zfs
/rpool              : zfs
/media/cdrom        : ufs
/media/cdrom-1      : ufs
/media/cdrom-2      : ufs
/media/cdrom-3      : ufs
/media/sol_10_811_sparc : hsfs
/media/cdrom-4      : ufs
/pond               : zfs
/pond/amy           : zfs
/pond/dr            : zfs
/pond/rory          : zfs
```

File System Monitoring Tool (fsstat)

A new file system monitoring tool, fsstat, is available to report file system operations from Oracle Solaris 11 release. You can use several options to report activity, such as by mount point or by file system type.

For example, the following fsstat command displays all ZFS file system operations since the ZFS module was loaded:

```
$ fsstat zfs
new  name   name  attr  attr lookup rddir  read read  write write
file remov  chng   get   set    ops   ops   ops bytes   ops bytes
268K  145K 93.6K 28.0M 71.1K   186M 2.74M 12.9M 56.2G 1.61M 9.46G zfs
```

For example, the following fsstat command displays all file system operations since the /export/ws file system mounted.

```
$ fsstat /export/ws
new  name   name  attr  attr lookup rddir  read read  write write
file remov  chng   get   set    ops   ops   ops bytes   ops bytes
0     0     0 18.1K     0  12.6M    52      0     0     0     0 /export/ws
```

The default form is to report statistical information in easy to understand values, such as GB, KB, and MB.

For more information, see fsstat(1M).

Where to Find File System Management Tasks

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

File System Management Task	For More Information
Connect and configure new disk devices.	Chapter 4, “Managing Disks in Oracle Solaris,” in “Managing Devices in Oracle Solaris 11.2 ”
Create and mount new file systems.	Chapter 2, “Creating and Mounting File Systems”
Make remote files available to users.	“Managing Network File Systems in Oracle Solaris 11.2 ”

♦ ♦ ♦ CHAPTER 2

Creating and Mounting File Systems

This chapter describes how to create and mount ZFS, temporary (TMPFS), and loopback (LOFS) file systems. Because TMPFS and LOFS are virtual file systems, you actually "access" them by mounting them. In addition, creating and mounting a legacy UFS file system is also covered.

This is a list of the information in this chapter:

- "Creating and Mounting Oracle Solaris File Systems" on page 23
- "Mounting and Unmounting Oracle Solaris File Systems" on page 27

Creating and Mounting Oracle Solaris File Systems

This section provides examples of creating and mounting Oracle Solaris file systems.

▼ How to Create an ZFS File System

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 ZFS storage pool.**

   The following example illustrates how to create a simple mirrored storage pool named `tank` and a ZFS file system named `tank` in one command. Assume that the whole disks `/dev/dsk/c1t0d0` and `/dev/dsk/c2t0d0` are available for use.

   ```
   # zpool create tank mirror c1t0d0 c2t0d0
   ```

3. **Create a ZFS file system.**

   ```
   # zfs create tank/fs
   ```

The new ZFS file system, `tank/fs`, can use as much of the disk space as needed, and is automatically mounted at `/tank/fs`.

4. **Confirm that the file system is created.**

```
# zfs list -r tank
NAME      USED  AVAIL  REFER  MOUNTPOINT
tank      117K   268G    21K  /tank
tank/fs    21K   268G    21K  /tank/fs
```

▼ How to Create and Mount a Legacy UFS File System

Before You Begin Ensure that you have met the following prerequisites:

- The disk must be formatted and divided into slices.
- If you are recreating an existing legacy UFS file system, unmount it.
- You need to know the device name of the slice that will contain the file system.

For information on finding disks and disk slice numbers, see Chapter 6, “Administering the System’s Disks,” in “Managing Devices in Oracle Solaris 11.2 ”.

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 legacy UFS file system.**

```
# newfs [-N] [-b size] [-i bytes] /dev/rdsk/device-name
```

   The system asks for confirmation.

Caution - Be sure you have specified the correct device name for the slice before performing this step. If you specify the wrong slice, you will erase its contents when the new file system is created. This error might cause the system to panic.

---

3. **To verify the creation of the legacy UFS file system, check the new file system.**

```
# fsck /dev/rdsk/device-name
```

   where *device-name* argument specifies the name of the disk device that contains the new file system.

The `fsck` command checks the consistency of the new file system, reports any problems, and prompts you before it repairs the problems. For more information on the `fsck` command, see `fsck`(1M).

4. **Mount the legacy UFS file system.**

```
# mkdir /directory-name
# mount /dev/dsk/device-name /directory-name
```

Example 2-1 Creating and Mounting a Legacy UFS File System

The following example shows how to create and mount a UFS file system `/dev/rdsk/c0t1d0s0` on `/legacy`.

```
# newfs /dev/rdsk/c0t1d0s0
newfs: construct a new file system /dev/rdsk/c0t1d0s0: (y/n)? y
/dev/rdsk/c0t1d0s0:     286722656 sectors in 46668 cylinders of 48 tracks, 128 sectors
140001.3MB in 2917 cyl groups (16 c/g, 48.00MB/g, 5824 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
32, 98464, 196896, 295328, 393760, 492192, 590624, 689056, 787488, 885920,
Initializing cylinder groups:
........................................................
super-block backups for last 10 cylinder groups at:
285773216, 285871648, 285970080, 286068512, 286166944, 286265376, 286363808,
286462240, 286560672, 286659104
# fsck /dev/rdsk/c0t1d0s0
# mkdir /legacy
# mount /dev/dsk/c0t1d0s0 /legacy
```

Next Steps To mount the legacy UFS file system automatically at boot time, after you create it, go to "How to Add an Entry to the `/etc/vfstab` File" on page 31.

How to Create and Mount a TMPFS File System

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 the directory that you want to mount as the TMPFS file system, if necessary.**

   ```
   # mkdir /mount-point
   ```

   where *mount-point* is the directory on which the TMPFS file system is mounted.

3. **Mount the TMPFS file system.**

```
# mount -F tmpfs [-o size=number]  swap mount-point
```

`-o size=`*number* Specifies the size limit of the TMPFS file system in MB.

mount-point Specifies the directory on which the TMPFS file system is mounted.

To set up the system to automatically mount a TMPFS file system at boot time, see Example 2-3.

4. **Verify that the TMPFS file system has been created.**

```
# mount -v
```

Example 2-2 Creating and Mounting a TMPFS File System

The following example shows how to create, mount, and limit the size of the TMPFS file system, `/export/reports`, to 50 MB.

```
# mkdir /export/reports
# chmod 777 /export/reports
# mount -F tmpfs -o size=50m swap /export/reports
# mount -v
```

Example 2-3 Mounting a TMPFS File System at Boot Time

You can set up the system to automatically mount a TMPFS file system at boot time by adding an `/etc/vfstab` entry. The following example shows an entry in the `/etc/vfstab` file that mounts `/export/test` as a TMPFS file system at boot time. Because the `size=`*number* option is not specified, the size of the TMPFS file system on `/export/test` is limited only by the available system resources.

```
swap - /export/test  tmpfs   -  yes  -
```

▼ How to Create and Mount an LOFS File System

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 the directory you want to mount as an LOFS file system, if necessary.**

```
# mkdir loopback-directory
```

3. **Grant the appropriate permissions and ownership on the newly created directory.**

4. **Create the mount point where you want to mount the LOFS file system, if necessary.**

   ```
   # mkdir /mount-point
   ```

5. **Mount the LOFS file system.**

   ```
   # mount -F lofs loopback-directory /mount-point
   ```

   loopback-directory Specifies the file system to be mounted on the loopback mount point.

   /mount-point Specifies the directory on which to mount the LOFS file system.

6. **Verify that the LOFS file system has been mounted.**

   ```
   # mount -v
   ```

Example 2-4 Creating and Mounting an LOFS File System

The following example shows how to create, mount, and test new software in the `/new/dist` directory as a loopback file system without actually having to install it.

```
# mkdir /tmp/newroot
# mount -F lofs /new/dist /tmp/newroot
# chroot /tmp/newroot newcommand
```

Example 2-5 Mounting an LOFS File System at Boot Time

You can set up the system to automatically mount an LOFS file system at boot time by adding an entry to the end of the `/etc/vfstab` file. The following example shows an entry in the `/etc/vfstab` file that mounts an LOFS file system for the root (`/`) file system on `/tmp/newroot`.

```
/ - /tmp/newroot  lofs   -  yes  -
```

Ensure that the loopback entries are the last entries in the `/etc/vfstab` file. Otherwise, if the `/etc/vfstab` entry for a loopback file system precedes the file systems to be included in it, the loopback file system cannot be mounted.

Mounting and Unmounting Oracle Solaris File Systems

ZFS file systems are mounted and unmounted automatically. You can make a legacy UFS file system available by mounting it, which attaches the file system to the system directory tree at the specified mount point. The root (`/`) file system is always mounted.

The following table provides guidelines on mounting file systems based on how you use them.

Mount Type Needed	Suggested Mount Method
Local or remote file systems that need to be mounted infrequently.	The mount command that you type manually from the command line.
Local legacy UFS file systems that need to be mounted frequently. Local ZFS file systems are automatically mounted by an SMF service.	The /etc/vfstab file, which mounts the file system automatically when the system is booted in multi user state.
Remote legacy UFS file systems, such as home directories, that need to be mounted frequently.	■ The /etc/vfstab file, which automatically mounts the file system when the system is booted in multiuser state. ■ autofs, which automatically mounts the file system when you access it or unmounts the file system when you change to another directory.

For more information on mounting removable media, see Chapter 14, “Managing Removable Media,” in “Managing Devices in Oracle Solaris 11.2 ”.

You can determine which file systems are already mounted by using the mount command:

```
$ mount [ -v ]
```

The -v displays the list of mounted file systems in verbose mode.

EXAMPLE 2-6 Determining Which File Systems Are Mounted

This example shows how to use the mount command to display information about the file systems that are currently mounted.

```
$ mount
/ on rpool/ROOT/zfsBE read/write/setuid/devices/rstchown/dev=3390002 on Tue ...
/devices on /devices read/write/setuid/devices/rstchown/dev=8580000 on Tue May 15 ...
/dev on /dev read/write/setuid/devices/rstchown/dev=85c0000 on Tue May 15 14:49:01 2012
/system/contract on ctfs read/write/setuid/devices/rstchown/dev=8680001 on Tue May 15 ...
/proc on proc read/write/setuid/devices/rstchown/dev=8600000 on Tue May 15 14:49:01 2012
/etc/mnttab on mnttab read/write/setuid/devices/rstchown/dev=86c0001 on Tue May 15 14:49:01 ...
/system/volatile on swap read/write/setuid/devices/rstchown/xattr/dev=8700001 on Tue May 15 ...
/system/object on objfs read/write/setuid/devices/rstchown/dev=8740001 on Tue May 15 ...
/etc/dfs/sharetab on sharefs read/write/setuid/devices/rstchown/dev=8780001 on Tue May 15 ...
/lib/libc.so.1 on /usr/lib/libc/libc_hwcap2.so.1 read/write/setuid/devices/rstchown/dev ...
/dev/fd on fd read/write/setuid/devices/rstchown/dev=8880001 on Tue May 15 14:49:29 2012
/var on rpool/ROOT/zfsBE/var read/write/setuid/devices/rstchown/nonbmand/exec/xattr/atime/ ...
/tmp on swap read/write/setuid/devices/rstchown/xattr/dev=8700002 on Tue May 15 14:49:29 2012
/var/share on rpool/VARSHARE read/write/setuid/devices/rstchown/nonbmand/exec/xattr/atime/ ...
/home/rimmer on pluto:/export/home/rimmer remote/read/write/setuid/xattr/...
```

This example shows how to use the zfs mount command to display information about ZFS file systems that are currently mounted.

```
$ zfs mount
rpool/ROOT/zfsBE                /
```

```
rpool/ROOT/zfsBE                /var
rpool/VARSHARE                  /var/share
tank/home                       /tank/home
```

Field Descriptions for the `/etc/vfstab` File

An entry in the `/etc/vfstab` file has seven fields, which are described in the following table.

TABLE 2-1 Field Descriptions for the `/etc/vfstab` File

Field Name	Description
`device to mount`	This field identifies one of the following: ■ The block device name for a local legacy UFS file system (for example, `/dev/dsk/c8t1d0s7`). ■ The resource name for a remote file system (for example, `myserver:/export/home`). After you add an entry for a remote system resource, be sure that the following service is enabled. `#` **`svcs -a | grep nfs/client`** `disabled       May_14   svc:/network/nfs/client:default` `#` **`svcadm enable svc:/network/nfs/client:default`** For more information about NFS, see "Managing Network File Systems in Oracle Solaris 11.2 ". ■ The swap volume (for example, `/dev/zvol/dsk/rpool/swap`). ■ A directory for a virtual file system.
`device to fsck`	The raw (character) device name that corresponds to the legacy UFS file system identified by the `device to mount` field (for example, `/dev/rdsk/c8t1d0s7`). This field determines the raw interface that is used by the `fsck` command. Use a dash (`-`) when there is no applicable device, such as for a read-only file system or a remote file system.
`mount point`	Identifies where to mount the legacy or remote file system (for example, `/data`).
`FS type`	Identifies the type of file system.
`fsck pass`	The pass number used by the `fsck` command to decide whether to check a legacy UFS file system. When the field contains a dash (`-`), the file system is not checked. Currently, the `fsck pass` value in the `/etc/vfstab` file is ignored during the boot process. When the field contains a zero, legacy UFS file systems are not checked. When the field contains a value greater than zero, the UFS file system is always checked. All legacy UFS file systems with a value of `1` in this field are checked one at a time in the order they appear in the `vfstab` file. When the `fsck` command is run on multiple UFS file systems that have `fsck pass` values greater than 1 and the preen option (`-o p`) is used, the `fsck` command automatically checks the file systems on different disks in parallel to maximize efficiency. Otherwise, the value of the pass number does not have any effect.
`mount at boot`	Set to `yes` or `no` for whether the file system should be automatically mounted by the `mountall` command when the system is booted. Note that this field has nothing to do with

Field Name	Description
	autofs. This field should always be set to `no` for virtual file systems such as `/proc` and `/dev/fd`.
`mount options`	A list of comma-separated options (with no spaces) that are used for mounting the file system. Use a dash (`-`) to indicate no options. For more information, see `vfstab`(4).

Note - You must have an entry in each field in the `/etc/vfstab` file. If there is no value for a field, be sure to specify a dash (`-`). Otherwise, the system might not boot successfully. Similarly, white space should not be used as a field value.

Prerequisites for Unmounting Oracle Solaris File Systems

The prerequisites for unmounting file systems include the following:

- You must be an administrator.
- You cannot unmount a file system that is busy. A file system is considered busy if a user is accessing a directory in the file system, if a program has a file open in that file system, or if the file system is being shared.

  You can make a file system available for unmounting by doing the following:

  - Changing to a directory in a different file system.
  - Logging out of the system.
  - Using the `fuser` command to list all processes that are accessing the file system and to stop them, if necessary. For more details, see "How to Stop All Processes That Are Accessing a File System" on page 35.

    Notify users if you need to unmount a file system that they are using.
  - Unsharing the file system.
    - For example:

      ```
      # zfs set share.nfs=off tank/fs1
      ```
    - Use the legacy unshare method. For information, see `unshare`(1M).

To verify that you unmounted a file system or a number of file systems, examine the output from the `mount` command:

```
$ mount | grep unmounted-file-system
```

▼ How to Add an Entry to the /etc/vfstab File

Use this procedure to mount non-ZFS file systems at boot time unless legacy mount behavior is needed for some ZFS file systems. For more information about mounting ZFS file systems, see "Managing ZFS File Systems in Oracle Solaris 11.2 ".

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 mount point for the file system to be mounted, if necessary.**

   ```
   # mkdir /mount-point
   ```

   There must be a mount point on the local system to mount a file system. A *mount point* is a directory to which the mounted file system is attached.

3. **Edit the /etc/vfstab file and add an entry. Ensure that you do the following:**

   a. **Separate each field with white space (a space or a tab).**

   b. **Specify a dash (-) if a field has no contents.**

   c. **Save the changes.**

> **Note -** Because the root (/) file system is mounted read-only by the kernel during the boot process, only the remount option (and options that can be used in conjunction with remount) affect the root (/) entry in the /etc/vfstab file.

Example 2-7 Adding an Entry to the /etc/vfstab File

The following example shows how to mount the disk slice /dev/dsk/c0t3d0s7 as a legacy UFS file system to the mount point /files1. The raw character device /dev/rdsk/c0t3d0s7 is specified as the device to fsck. The fsck pass value of 2 means that the file system will be checked, but not sequentially.

```
#device           device             mount    FS       fsck   mount    mount
#to mount         to fsck            point    type     pass   at boot  options
#
/dev/dsk/c0t3d0s7 /dev/rdsk/c0t3d0s7 /files1  ufs      2      yes       -
```

The following example shows how to mount the legacy /export/man directory from the system pluto as an NFS file system on mount point /usr/man. Neither a device to fsck nor a fsck pass is specified because it's an NFS file system. In this example, mount options are ro (read-only) and soft.

```
#device           device             mount    FS       fsck   mount    mount
#to mount         to fsck            point    type     pass   at boot  options
pluto:/export/man   -                /usr/man nfs      -      yes       ro,soft
```

After you add the remote system and resource to the /etc/vfstab file, be sure that the following service is started.

```
# svcs -a | grep nfs/client
disabled       May_14   svc:/network/nfs/client:default
# svcadm enable svc:/network/nfs/client:default
```

Otherwise, the remote file system will not be mounted after the system is rebooted.

▼ How to Mount a File System (/etc/vfstab File)

1. **Become an administrator.**

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

2. **Mount a file system listed in the /etc/vfstab file.**

   ```
   # mount /mount-point
   ```

   where */mount-point* specifies an entry in the mount point or device to mount field in the /etc/vfstab file. It is usually easier to specify the mount point.

Example 2-8 Mounting a File System (/etc/vfstab File)

The following example shows how to mount the local /legacy file system that is listed in the /etc/vfstab file.

```
# mount /legacy
```

Example 2-9 Mounting All File Systems (/etc/vfstab File)

The following example shows the messages that are displayed when you use the mountall command and the file systems are already mounted.

```
# mountall
mount: /tmp is already mounted or swap is busy
```

The following example shows how to mount all the local systems that are listed in the /etc/vfstab file.

```
# mountall -l
```

The following example shows how to mount all available ZFS file systems.

```
# zfs mount -a
```

The following example shows how to mount all the remote file systems that are listed in the /etc/vfstab file.

```
# mountall -r
```

▼ How to Mount an NFS File System (mount Command)

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 mount point for the file system to be mounted, if necessary.**

   ```
   # mkdir /mount-point
   ```

   There must be a mount point on the local system to mount a file system. A mount point is a directory to which the mounted file system is attached.

3. **Ensure that the resource (file or directory) is available from a server.**

   To mount an NFS file system, the resource must be made available on the server by using the `share` command. For information on how to share resources, see "Managing Network File Systems in Oracle Solaris 11.2".

4. **Mount the NFS file system.**

   ```
   # mount -F nfs [-o mount-options] server:/directory /mount-point
   ```

Example 2-10 Mounting an NFS File System (`mount` Command)

The following example shows how to mount the `/export/packages` directory on `/mnt` from the server `pluto`.

```
# mount -F nfs pluto:/export/packages /mnt
```

▼ x86: How to Mount a PCFS (DOS) File System From a Hard Disk (`mount` Command)

Use the following procedure to mount a PCFS (DOS) file system from a hard disk.

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 mount point for the file system to be mounted, if necessary.**

   ```
   # mkdir /mount-point
   ```

   There must be a mount point on the local system to mount a file system. A *mount point* is a directory to which the mounted file system is attached.

3. **Mount the PCFS file system.**

   ```
   # mount -F pcfs [-o rw | ro] /dev/dsk/device-name:logical-drive /mount-point
   ```

   `-o rw | ro` Specifies that you can mount a PCFS file system read/write (`rw`) or read-only (`ro`). If you do not specify this option, the default is `rw`.

   `/dev/dsk/`*device-name* Specifies the device name of the whole disk (for example, `/dev/dsk/c0t0d0p0`).

   logical-drive Specifies either the DOS logical drive letter (c through z) or a drive number (1 through 24). Drive c is equivalent to drive 1 and represents the primary DOS slice on the drive. All other letters or numbers represent DOS logical drives within the extended DOS slice.

   /mount-point Specifies the directory on which to mount the file system.

   Note that the *device-name* and *logical-drive* must be separated by a colon.

Example 2-11 x86: Mounting a PCFS (DOS) File System From a Hard Disk (`mount` Command)

The following example shows how to mount the logical drive in the primary DOS slice on the `/pcfs/c` directory.

```
# mount -F pcfs /dev/dsk/c0t0d0p0:c /pcfs/c
```

The following example shows how to mount read-only the first logical drive in the extended DOS slice on the `/mnt` directory.

```
# mount -F pcfs -o ro /dev/dsk/c0t0d0p0:2 /mnt
```

▼ How to Stop All Processes That Are Accessing a File System

1. **Become an administrator.**

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

2. **List all the processes that are accessing the file system so that you know which processes you are going to stop.**

   ```
   # fuser -c [ -u ] /mount-point
   ```

   `-c` Reports on files that are mount points for file systems and any files within those mounted file systems.

   `-u` Displays the user login name for each process ID.

   /mount-point Specifies the name of the file system for which you want to stop processes.

3. **Stop all processes that are accessing the file system.**

   ```
   # fuser -c -k /mount-point
   ```

   A `SIGKILL` is sent to each process that is using the file system.

   Note - You should not stop a user's processes without first warning the user.

   ---

4. **Verify that no processes are accessing the file system.**

   ```
   # fuser -c /mount-point
   ```

Example 2-12 Stopping All Processes That Are Accessing a File System

The following example shows how to stop process `4006c` that is using the `/export/home` file system.

```
# fuser -c /export/home
/export/home:     4006c
# fuser -c -k /export/home
/export/home:     4006c
# fuser -c /export/home
/export/home:
```

▼ How to Unmount a File System

Use the following procedure to unmount a file system.

1. **Ensure that you have met the prerequisites listed in "Prerequisites for Unmounting Oracle Solaris File Systems" on page 30.**

2. **Unmount the file system.**

   ```
   # umount /mount-point
   ```

   where */mount-point* is the name of the file system that you want to unmount.

   This name can be one of the following:

   - The directory name where the file system is mounted
   - The device name path of the file system
   - The resource for an NFS file system
   - The loopback directory for an LOFS file system

Example 2-13 Unmounting a File System

The following example shows how to unmount a legacy UFS file system:

```
# umount /legacy
```

The following example shows how to forcibly unmount the UFS `/legacy` file system:

```
# umount -f /legacy
```

The following example shows to unmount all ZFS file systems:

```
# zfs umount -a
```

All file systems are unmounted, except for those file systems that are busy.

♦ ♦ ♦ CHAPTER 3

Configuring Additional Swap Space

This chapter provides guidelines and step-by-step instructions for configuring additional swap space for a ZFS root file system after the Oracle Solaris OS is installed.

This is a list of the information in this chapter:

About Swap Space

You should understand the features of the swap mechanism in Oracle Solaris to determine the following:

- Swap space requirements
- The relationship between swap space and the TMPFS file system
- How to recover from error messages related to swap space

Swap Space and Virtual Memory

Oracle Solaris OS software and application software can use some disk space for temporary storage rather than for file systems. The reserved area of the disk is called *swap* space. Swap space is used as virtual memory storage areas when the system does not have enough physical memory to handle current processes. In a ZFS root file system, the disk space reserved for swap is a ZFS volume.

The virtual memory system maps physical copies of files on disk to virtual addresses in memory. Physical memory pages that contain the data for these mappings can be backed by regular files in the file system, or by swap space. If the memory is backed by swap space it is referred to as *anonymous memory* because no identity is assigned to the disk space that is backing the memory.

The Oracle Solaris OS uses the concept of *virtual swap space*, a layer between anonymous memory pages and the physical storage (or disk-backed swap space) that actually back these pages. A system's virtual swap space is equal to the sum of all its physical (disk-backed) swap space plus a portion of the currently available physical memory.

Virtual swap space has these advantages:

- The need for large amounts of physical swap space is reduced because virtual swap space does not necessarily correspond to physical (disk) storage.
- A pseudo file system called SWAPFS provides addresses for anonymous memory pages. Because SWAPFS controls the allocation of memory pages, it has greater flexibility in deciding what happens to a page. For example, SWAPFS might change the page's requirements for disk-backed swap storage.

Swap Space and the TMPFS File System

The TMPFS file system is activated automatically in the Oracle Solaris environment by an entry in the `/etc/vfstab` file. The TMPFS file system stores files and their associated information in memory (in the `/tmp` directory) rather than on disk, which speeds access to those files. This feature results in a major performance enhancement for applications such as compilers and DBMS products that use `/tmp` heavily.

The TMPFS file system allocates space in the `/tmp` directory from the system's swap resources. This feature means that as you use up space in the `/tmp` directory, you are also using up swap space. So, if your applications use the `/tmp` directory heavily and you do not monitor swap space usage, your system could run out of swap space.

Do use the following if you want to use TMPFS, but your swap resources are limited:

- Mount the TMPFS file system with the size option (`-o` *size*) to control how much swap resources TMPFS can use.
- Use your compiler's `TMPDIR` environment variable to point to another larger directory.

  Using your compiler's `TMPDIR` variable only controls whether the compiler is using the `/tmp` directory. This variable has no effect on other programs' use of the `/tmp` directory.

Swap Space and Dump Device Configuration

A *dump device* is usually disk space that is reserved to store system crash dump information. When a system is installed, a ZFS swap volume and dump volume are created automatically.

You can change a system's dump volume by using the `dumpadm` command. For more information, see "Troubleshooting System Administration Issues in Oracle Solaris 11.2 ".

In a ZFS root environment, swap and dump are configured as separate ZFS volumes. The advantages to this model are as follows:

- You don't have to partition a disk to include swap and dump areas.
- Swap and dump devices benefit from the underlying ZFS I/O pipeline architecture.
- You can set characteristics, such as compression, on swap and dump devices.
- You can reset both swap and dump device sizes. For example:

  ```
  # zfs set volsize=2G rpool/dump
  # zfs get volsize rpool/dump
  NAME        PROPERTY  VALUE       SOURCE
  rpool/dump  volsize   2G          -
  ```

  Keep in mind that reallocating a large dump device is a time-consuming process.
- On an active system, you might consider adding a second swap volume to increase overall swap size.

For more information about using ZFS swap and dump devices, see "Managing Your ZFS Swap and Dump Devices" in "Managing ZFS File Systems in Oracle Solaris 11.2 ".

Swap Space and Dynamic Reconfiguration

A good practice is to allocate enough swap space to support a failing CPU or system board during dynamic reconfiguration. Otherwise, a CPU or system board failure might result in your host or domain rebooting with less memory.

Without having this additional swap space available, one or more of your applications might fail to start due to insufficient memory. This problem would require manual intervention either to add additional swap space or to reconfigure the memory usage of these applications.

If you have allocated additional swap space to handle a potential loss of memory on reboot, all of your intensive applications might start as usual. This means the system will be available to the users, perhaps possibly slower due to some additional swapping.

For more information, see your hardware dynamic reconfiguration guide.

Configuring Swap Space in a SAN Environment

Review the following points to determine whether you might configure swap space on a network-connected disk, such as in a SAN environment:

- Diagnosing swap space issues on a locally-attached disk is easier than diagnosing swap space issues on a network-connected disk.
- The performance of swap space over a SAN should be comparable to swap space configured on a locally-attached disk.
- Adding more memory to a system with performance issues, after analyzing performance data, might resolve a swap over SAN performance problem better than moving the swap to a locally-attached disk.

How Do I Know If I Need More Swap Space?

Use the `swap -l` command to determine if your system needs more swap space.

For example, the following `swap -l` output shows that this system's swap space is almost entirely consumed or at 100% allocation.

```
$ swap -l
swapfile                   dev    swaplo   blocks     free
/dev/zvol/dsk/rpool/swap  182,2       16   67108848     92
```

When a system's swap space is at 100% allocation, an application's memory pages become temporarily locked. Application errors might not occur, but system performance will likely suffer.

Swap-Related Error Messages

These messages indicate that an application was trying to get more anonymous memory. However, no swap space was left to back it.

```
application is out of memory
```

```
malloc error O
```

```
messages.1:Sep 21 20:52:11 mars genunix: [ID 470503 kern.warning]
WARNING: Sorry, no swap space to grow stack for pid 100295 (myprog)
```

TMPFS-Related Error Messages

The following message is displayed if a page could not be allocated when a file was being written. This problem can occur when TMPFS tries to write more than it is allowed or if currently executed programs are using a lot of memory.

```
directory: File system full, swap space limit exceeded
```

The following message means that TMPFS ran out of physical memory while attempting to create a new file or directory:

```
directory: File system full, memory allocation failed
```

For information on recovering from the TMPFS-related error messages, see `tmpfs`(7FS).

How Swap Space Is Allocated

Initially, swap space is allocated as part of the Oracle Solaris installation process and the swap space size varies based on system memory.

For general guidelines on allocating swap space, see "Planning for Swap Space" on page 41.

Swap Areas and the `/etc/vfstab` File

After the system is installed, swap areas and swap files are listed in the `/etc/vfstab` file. They are activated by the `/sbin/swapadd` script when the system is booted.

An entry for a swap device in the `/etc/vfstab` file contains the following:

- The full path name of the swap volume path name on a system with a ZFS root file system
- File system type of the swap slice or swap file

The file system that contains a swap file must be mounted before the swap file is activated. So, in the `/etc/vfstab` file, ensure that the entry that mounts the file system comes before the entry that activates the swap file.

Planning for Swap Space

The most important factors in determining swap space size are the requirements of the system's software applications. For example, large applications such as computer-aided design simulators, database management products, transaction monitors, and geologic analysis systems can consume as much as 200–1000 MB of swap space.

Consult your application vendors for swap space requirements for their applications.

If you are unable to determine swap space requirements from your application vendors, use the following general guidelines based on your system type to allocate swap space.

Note - Crash dump content is compressed so the dump device does not have to be the same size as physical memory. By default, the dump content value is set to kernel pages. However, if the dump content value is set to dump all memory pages, then consider increasing the dump size to half the size of physical memory or more.

TABLE 3-1 Swap and Dump Volume Sizes for ZFS File Systems

System Type	Swap Volume Size	Dump Volume Size
System with about 4 GB of physical memory	1 GB	2 GB
Mid-range server with about 8 GB of physical memory	2 GB	4 GB
High-end server with about 16 to 128 GB of physical memory	4 GB	8-64 GB
High-end server with more than 128 GB of physical memory	1/4 of physical memory size	1/2 of physical memory size

Note - A busy system with many active ZFS file systems might use 1/2 to 3/4 the size of physical memory for the size of the dump device.

On a system with a ZFS root file system, if you attempt to designate a dump device that is too small to hold a system crash system with the `dumpadm -d` command, you will see a message similar to the following:

```
dumpadm: dump device /dev/zvol/dsk/rpool/dump is too small to hold a
system dump dump size 43467329536 bytes, device size 42949672960 bytes
```

Allocating Swap Space for ZFS-Based Systems

During an initial installation of a ZFS root file system, a swap area is automatically created on a ZFS volume in the ZFS root pool.

- Swap devices are not pre-allocated to fixed-size slices, so it is fairly easy to modify the swap size later.
- After you assess the swap requirements of your applications, you can use the default swap size or adjust the swap volume size during an initial installation or after the installation, if necessary.
- In a ZFS environment, file systems consume space from the pool so the `/var/crash` directory consumes what it needs depending on how many crash dumps are saved.

Consider the following issues when creating swap volumes:

- Separate ZFS volumes must be used for the swap area and dump devices.
- Currently, using a swap file on a ZFS file system is not supported.
- If you want to create swap and dump devices in a non-root pool, do not create swap and dump volumes in a RAID-Z pool. If a pool includes swap and dump volumes, it must be a one-disk pool or a mirrored pool. Otherwise, you will see a message similar to the following:

  ```
  /dev/zvol/dsk/rzpool/swap: Operation not supported
  ```

Monitoring Swap Resources

The /usr/sbin/swap command is used to manage swap areas. Two options, -l and -s, display information about swap resources.

Use the swap -l command to identify a system's swap areas. Activated swap devices or files are listed under the swapfile column. For example:

```
# swap -l
swapfile             dev  swaplo blocks   free
/dev/dsk/c0t0d0s1   136,1      16 1638608 1600528
```

On a system with a ZFS root file system, the swap -l command identifies similar output except that it identifies the ZFS volume path name. For example:

```
# swap -l
swapfile                  dev  swaplo blocks   free
/dev/zvol/dsk/rpool/swap 256,1      16 1058800 1058800
```

Use the swap -s command to monitor swap resources.

```
# swap -s
total: 57416k bytes allocated + 10480k reserved = 67896k used,
833128k available
```

The used value plus the available value equals the total swap space on the system, which includes a portion of physical memory and swap devices (or files).

You can use the amount of available and used swap space (in the swap -s output) as a way to monitor swap space usage over time. If a system's performance is good, use swap -s to determine how much swap space is available. When the performance of a system slows down, check the amount of available swap space to determine if it has decreased. Then you can identify what changes to the system might have caused swap space usage to increase.

When using this command, keep in mind that the amount of physical memory available for swap usage changes dynamically as the kernel and user processes lock down and release physical memory.

Note - The `swap -l` command displays swap space in 512-byte blocks. The `swap -s` command displays swap space in 1024-byte blocks. If you add up the blocks from `swap -l` and convert them to KB, the result is less than `used` + `available` (in the `swap -s` output). The reason is that `swap -l` does not include physical memory in its calculation of swap space.

The output from the `swap -s` command is summarized in the following table.

TABLE 3-2 Output of the `swap -s` Command

Keyword	Description
`bytes allocated`	The total amount of swap space in 1024-byte blocks that is currently allocated as backing store (disk-backed swap space).
`reserved`	The total amount of swap space in 1024-byte blocks that is not currently allocated, but claimed by memory for possible future use.
`used`	The total amount of swap space in 1024-byte blocks that is either allocated or reserved.
`available`	The total amount of swap space in 1024-byte blocks that is currently available for future reservation and allocation.

Adding or Changing Swap Space in an Oracle Solaris ZFS Root Environment

The following section describes how to add or change swap space in a ZFS root environment. See the previous sections to determine if your system or applications need additional swap space.

For more information about changing swap and dump volumes in a ZFS root environment, see “Managing Your ZFS Swap and Dump Devices” in “Managing ZFS File Systems in Oracle Solaris 11.2 ”.

▼ How to Add Swap Space in an Oracle Solaris ZFS Root Environment

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 current swap volume.**

   A swap volume cannot be removed if it is in use. You can tell if the current swap volume is in use by comparing the blocks identified in the blocks column and blocks identified in the free column. If the blocks in the two columns are equal, the swap area is not busy. For example:

   ```
   # swap -l
   swapfile                 dev  swaplo   blocks   free
   /dev/zvol/dsk/rpool/swap 256,1      16 1058800 1058800
   ```

3. **Select one of the following to resize the swap volume.**

   a. **If the current swap area is not in use, you can resize the size of the current swap volume, but you must reboot the system to see the increased swap space.**

      For example:

      ```
      # zfs get volsize rpool/swap
      NAME        PROPERTY  VALUE    SOURCE
      rpool/swap  volsize   517M     -
      # zfs set volsize=2g rpool/swap
      # zfs get volsize rpool/swap
      NAME        PROPERTY  VALUE    SOURCE
      rpool/swap  volsize   2G       -
      # init 6
      ```

   b. **If the system cannot be rebooted, add another swap volume to increase your total swap space.**

      For example:

      ```
      # zfs create -V 2G rpool/swap2
      ```

      Then, activate the second swap volume.

      ```
      # swap -a /dev/zvol/dsk/rpool/swap2
      # swap -l
      swapfile                  dev  swaplo   blocks   free
      /dev/zvol/dsk/rpool/swap  256,1      16 1058800 1058800
      /dev/zvol/dsk/rpool/swap2 256,3      16 4194288 4194288
      ```

4. **If necessary, add an entry for the second swap volume in the `/etc/vfstab` file.**

   For example:

   ```
   /dev/zvol/dsk/rpool/swap2    -        -       swap    -       no      -
   ```

♦ ♦ ♦ CHAPTER 4

Copying Files and File Systems

This chapter describes how to copy files and file systems to disk, tape, and diskettes by using various backup commands.

This is a list of the step-by-step instructions in this chapter.

Commands for Copying File Systems

When you want to copy or move individual files, portions of file systems, or complete file systems, you can use the procedures described in this chapter.

The following table describes various backup and restore commands that are available in the Oracle Solaris release. For enterprise environments, consider using a enterprise-level backup product. Information about enterprise-level backup products is available on the Oracle Technical Network.

TABLE 4-1 Summary of Various Backup Commands

Command Name	Aware of File System Boundaries?	Supports Multiple Volume Backups?	Physical or Logical Copy?
volcopy	Yes	Yes	Physical

Command Name	Aware of File System Boundaries?	Supports Multiple Volume Backups?	Physical or Logical Copy?
`tar`	No	No	Logical
`cpio`	No	Yes	Logical
`pax`	Yes	Yes	Logical
`dd`	Yes	No	Physical
`zfs send` and `zfs receive`	Yes	N/A	Logical
`zfs snapshot`	Yes	N/A	Logical

The following table describes the advantages and disadvantages of some of these commands.

TABLE 4-2 Advantages and Disadvantages of `tar`, `pax`, and `cpio` Commands

Command	Function	Advantages	Disadvantages
`tar`	Use to copy files and directory subtrees to a single tape.	■ Available on most UNIX operating systems ■ Public domain versions are readily available	■ Is not aware of file system boundaries ■ Length of full path name cannot exceed 255 characters ■ Cannot be used to create multiple tape volumes
`pax`	Use to copy files, special files, or file systems that require multiple tape volumes. Or, use when you want to copy files to and from POSIX-compliant systems.	■ Better portability than the `tar` or `cpio` commands for POSIX-compliant systems ■ Multiple vendor support	Same disadvantages as the `tar` command, except that the `pax` command can create multiple tape volumes.
`cpio`	Use to copy files, special files, or file systems that require multiple tape volumes. Or, use when you want to copy files from systems running current Oracle Solaris releases systems to systems running older Solaris releases.	■ Packs data onto tape more efficiently than the `tar` command ■ Skips over any bad spots in a tape when restoring ■ Provides options for writing files with different header formats, such as (`tar`, `ustar`, `crc`, `odc`, `bar`), for portability between different system types ■ Creates multiple tape volumes	The command syntax is more difficult than the `tar` or `pax` commands.

The following sections describes step-by-step instructions and examples of how to use these commands.

Copying Directories Between File Systems (`cpio` Command)

You can use the `cpio` (copy in and out) command to copy individual files, groups of files, or complete file systems. This section describes how to use the `cpio` command to copy complete file systems.

The `cpio` command is an archiving program that copies a list of files into a single, large output file. This command inserts headers between the individual files to facilitate recovery. You can use the `cpio` command to copy complete file systems to another slice, another system, or to a media device, such as a tape or USB diskette.

Because the `cpio` command recognizes end-of-media and prompts you to insert another volume, it is the most effective command to use to create archives that require multiple tapes or USB diskettes.

With the `cpio` command, you frequently use the `ls` and `find` commands to list and select the files you want to copy, and then to pipe the output to the `cpio` command.

▼ How to Copy Directories Between File Systems (`cpio`)

1. **Become an administrator.**

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

2. **Change to the appropriate directory.**

   ```
   # cd filesystem1
   ```

3. **Copy the directory tree from *filesystem1* to *filesystem2* by using a combination of the `find` and `cpio` commands.**

   ```
   # find . -print -depth | cpio -pdm filesystem2
   ```

   `.` Starts in the current working directory.

   `-print` Prints the file names.

   `-depth` Descends the directory hierarchy and prints file names from the bottom up.

   `-p` Creates a list of files.

`-d`	Creates directories as needed.
`-m`	Sets the correct modification times on directories.

For more information, see `cpio`(1).

The files from the directory name you specify are copied. The symbolic links are preserved.

You might also specify the `-u` option. This option forces an unconditional copy. Otherwise, older files do not replace newer files. This option might be useful if you want an exact copy of a directory, and some of the files being copied might already exist in the target directory.

4. **Verify that the copy was successful by displaying the contents of the destination directory.**

```
# cd filesystem2
# ls
```

5. **If appropriate, remove the source directory.**

```
# rm -rf filesystem1
```

Example 4-1 Copying Directories Between File Systems (`cpio`)

```
# cd /data1
# find . -print -depth | cpio -pdm /data2
19013 blocks
# cd /data2
# ls
# rm -rf /data1
```

Copying Files and File Systems to Tape

You can use the `tar`, `pax`, and `cpio` commands to copy files and file systems to tape. The command that you choose depends on how much flexibility and precision you require for the copy. Because all three commands use the raw device, you do not need to format or make a file system on tapes before you use them.

The tape drive and device name that you use depend on the hardware configuration for each system. For more information about tape device names, see "Choosing Which Media to Use" in "Managing Devices in Oracle Solaris 11.2 ".

Copying Files to Tape (tar Command)

Here is information that you should know before you copy files to tape with the tar command:

- Copying files to a tape with the -c option to the tar command destroys any files already on the tape at or beyond the current tape position.
- You can use file name substitution wildcards (? and *) as part of the file names that you specify when copying files. For example, to copy all documents with a .doc suffix, type *.doc as the file name argument.
- You cannot use file name substitution wildcards when you extract files from a tar archive.

▼ How to Copy Files to a Tape (tar)

1. **Change to the directory that contains the files you want to copy.**
2. **Insert a write-enabled tape into the tape drive.**
3. **Copy the files to tape.**

```
$ tar cvf /dev/rmt/n filenames
```

c	Indicates that you want to create an archive.
v	Displays the name of each file as it is archived.
f /dev/rmt/*n*	Indicates that the archive should be written to the specified device or file.
filenames	Indicates the files and directories that you want to copy. Separate multiple files with spaces.

The file names that you specify are copied to the tape, overwriting any existing files on the tape.

4. **Remove the tape from the drive. Write the names of the files on the tape label.**
5. **Verify that the files you copied are on the tape.**

```
$ tar tvf /dev/rmt/n
```

For more information on listing files on a tar tape, see "How to List the Files on a Tape (tar)" on page 52.

Example 4-2 Copying Files to a Tape (tar)

The following example shows how to copy three files to the tape in tape drive 0.

```
$ cd /export/home/kryten
$ ls reports
reportA reportB reportC
$ tar cvf /dev/rmt/0 reports
a reports/ 0 tape blocks
a reports/reportA 59 tape blocks
a reports/reportB 61 tape blocks
a reports/reportC 63 tape blocks
$ tar tvf /dev/rmt/0
```

How to List the Files on a Tape (`tar`)

1. **Insert a tape into the tape drive.**

2. **Display the tape contents.**

```
$ tar tvf /dev/rmt/n
```

`t` Lists the table of contents for the files on the tape.

`v` Used with the `t` option, and provides detailed information about the files on the tape.

`f /dev/rmt/`*n* Indicates the tape device.

Example 4-3 Listing the Files on a Tape (`tar`)

The following example shows a listing of files on the tape in drive 0.

```
$ tar tvf /dev/rmt/0
drwxr-xr-x   0/0        0 Jul 14 13:50 2010 reports/
-r--r--r--   0/0   206663 Jul 14 13:50 2010 reports/reportC
-r--r--r--   0/0   206663 Jul 14 13:50 2010 reports/reportB
-r--r--r--   0/0   206663 Jul 14 13:50 2010 reports/reportA
```

How to Retrieve Files From a Tape (`tar`)

1. **Change to the directory where you want to put the files.**

2. **Insert the tape into the tape drive.**

3. **Retrieve the files from the tape.**

```
$ tar xvf /dev/rmt/n [filenames]
```

`x` Indicates that the files should be extracted from the specified archive file. All files on the tape in the specified drive are copied to the current directory.

`v` Displays the name of each file as it is retrieved.

`f /dev/rmt/`*n* Indicates the tape device that contains the archive.

filenames Specifies a file to retrieve. Separate multiple files with spaces.

For more information, see `tar`(1).

4. **Verify that the files have been copied.**

```
$ ls -l
```

Example 4-4 Retrieving Files on a Tape (`tar`)

The following example shows how to retrieve all the files from the tape in drive 0.

```
$ cd /var/tmp
$ tar xvf /dev/rmt/0
x reports/, 0 bytes, 0 tape blocks
x reports/reportA, 0 bytes, 0 tape blocks
x reports/reportB, 0 bytes, 0 tape blocks
x reports/reportC, 0 bytes, 0 tape blocks
x reports/reportD, 0 bytes, 0 tape blocks
$ ls -l
```

Troubleshooting The names of the files extracted from the tape must exactly match the names of the files that are stored on the archive. If you have any doubts about the names or paths of the files, first list the files on the tape. For instructions on listing the files on the tape, see "How to List the Files on a Tape (`tar`)" on page 52.

Copying Files to a Tape With the `pax` Command

This section describes how to copy files to a tape with the `pax` command.

How to Copy Files to a Tape (`pax`)

1. **Change to the directory that contains the files you want to copy.**

2. **Insert a write-enabled tape into the tape drive.**

3. **Copy the files to tape.**

   ```
   $ pax -w -f /dev/rmt/n filenames
   ```

   `-w` Enables the write mode.

   `-f /dev/rmt/n` Identifies the tape drive.

   filenames Indicates the files and directories that you want to copy. Separate multiple files with spaces.

   For more information, see `pax(1)`.

4. **Verify that the files have been copied to tape.**

   ```
   $ pax -f /dev/rmt/n
   ```

5. **Remove the tape from the drive. Write the names of the files on the tape label.**

Example 4-5 Copying Files to a Tape (`pax`)

The following example shows how to use the `pax` command to copy all the files in the current directory.

```
$ pax -w -f /dev/rmt/0 .
$ pax -f /dev/rmt/0
filea fileb filec
```

Copying Files to Tape With the `cpio` Command

This section describes how to copy files to tape with the `cpio` command.

How to Copy All Files in a Directory to a Tape (`cpio`)

1. **Change to the directory that contains the files you want to copy.**

2. **Insert a write-enabled tape into the tape drive.**

3. **Copy the files to tape.**

   ```
   $ ls | cpio -oc > /dev/rmt/n
   ```

`ls` — Provides the `cpio` command with a list of file names.

`cpio -oc` — Specifies that the `cpio` command should operate in copy-out mode (`-o`) and write header information in ASCII character format (`-c`). These options ensure portability to other vendors' systems.

`> /dev/rmt/`*n* — Specifies the output file.

All files in the directory are copied to the tape in the drive you specify, overwriting any existing files on the tape. The total number of blocks that are copied is shown.

4. **Verify that the files have been copied to tape.**

```
$ cpio -civt < /dev/rmt/n
```

`-c` — Specifies that the `cpio` command should read files in ASCII character format.

`-i` — Specifies that the `cpio` command should operate in copy-in mode, even though the command is only listing files at this point.

`-v` — Displays the output in a format that is similar to the output from the `ls -l` command.

`-t` — Lists the table of contents for the files on the tape in the tape drive that you specify.

`< /dev/rmt/`*n* — Specifies the input file of an existing `cpio` archive.

5. **Remove the tape from the drive. Write the names of the files on the tape label.**

Example 4-6 Copying All Files in a Directory to a Tape (`cpio`)

The following example shows how to copy all of the files in the `/export/home/kryten` directory to the tape in tape drive 0.

```
$ cd /export/home/kryten
$ ls | cpio -oc > /dev/rmt/0
1280 blocks
$ cpio -civt < /dev/rmt/0
-r--r--r--    1 kryten   staff      206663 Jul 14 13:52 2010, filea
-r--r--r--    1 kryten   staff      206663 Jul 14 13:52 2010, fileb
-r--r--r--    1 kryten   staff      206663 Jul 14 13:52 2010, filec
drwxr-xr-x    2 kryten   staff           0 Jul 14 13:52 2010, letters
drwxr-xr-x    2 kryten   staff           0 Jul 14 13:52 2010, reports
1280 blocks
```

How to List the Files on a Tape (`cpio`)

Note - Listing the table of contents on a tape takes a long time because the `cpio` command must process the entire archive.

---

1. **Insert an archive tape into the tape drive.**
2. **List the files on the tape.**

```
$ cpio -civt < /dev/rmt/n
```

Example 4-7 Listing the Files on a Tape (`cpio`)

The following example shows how to list the files on the tape in drive 0.

```
$ cpio -civt < /dev/rmt/0
-r--r--r--    1 kryten   staff      206663 Jul 14 13:52 2010, filea
-r--r--r--    1 kryten   staff      206663 Jul 14 13:52 2010, fileb
-r--r--r--    1 kryten   staff      206663 Jul 14 13:52 2010, filec
drwxr-xr-x    2 kryten   staff           0 Jul 14 13:52 2010, letters
drwxr-xr-x    2 kryten   staff           0 Jul 14 13:52 2010, reports
1280 blocks
```

How to Retrieve All Files From a Tape (`cpio`)

If the archive was created using relative path names, the input files are built as a directory within the current directory when you retrieve the files. If, however, the archive was created with absolute path names, the same absolute paths are used to recreate the file on your system.

Caution - The use of absolute path names can be dangerous because you might overwrite existing files on your system.

---

1. **Change to the directory where you want to put the files.**
2. **Insert the tape into the tape drive.**
3. **Extract all files from the tape.**

```
$ cpio -icvd < /dev/rmt/n
```

`-i` Extracts files from standard input.

`-c` Specifies that the `cpio` command should read files in ASCII character format.

`-v` Displays the files as they are retrieved in a format that is similar to the output from the `ls` command.

`-d` Creates directories as needed.

`< /dev/rmt/`*n* Specifies the output file.

4. **Verify that the files were copied.**

```
$ ls -l
```

Example 4-8 Retrieving All Files From a Tape (`cpio`)

The following example shows how to retrieve all files from the tape in drive 0.

```
$ cd /var/tmp
cpio -icvd < /dev/rmt/0
answers
sc.directives
tests
8 blocks
$ ls -l
```

How to Retrieve Specific Files From a Tape (`cpio`)

1. **Change to the directory where you want to put the files.**
2. **Insert the tape into the tape drive.**
3. **Retrieve a subset of files from the tape.**

```
$ cpio -icv "*file" < /dev/rmt/n
```

`-i` Extracts files from standard input.

`-c` Specifies that the `cpio` command should read headers in ASCII character format.

`-v` Displays the files as they are retrieved in a format that is similar to the output from the `ls` command.

"*file" — Specifies that all files that match the pattern are copied to the current directory. You can specify multiple patterns, but each pattern must be enclosed in double quotation marks.

< /dev/rmt/n — Specifies the input file.

For more information, see cpio(1).

4. **Verify that the files were copied.**

```
$ ls -l
```

Example 4-9 Retrieving Specific Files From a Tape (cpio)

The following example shows how to retrieve all files with the chapter suffix from the tape in drive 0.

```
$ cd /home/smith/Book
$ cpio -icv "*chapter" < /dev/rmt/0
Boot.chapter
Directory.chapter
Install.chapter
Intro.chapter
31 blocks
$ ls -l
```

Copying Files to a Remote Tape Device

This section describes how to copy files to a remote tape device using tar and dd command.

How to Copy Files to a Remote Tape Device (tar and dd)

1. **Configure ssh on the remote system so that you can access the tape drive. See "Configuring Secure Shell (Tasks)" in "Managing Secure Shell Access in Oracle Solaris 11.2 ".**

2. **Change to the directory where you want to put the files.**

3. **Insert the tape into the tape drive.**

4. **Copy the files to a remote tape drive.**

```
$ tar cvf - filenames | ssh remote-host dd of=/dev/rmt/n obs=block-size
```

`tar cf`	Creates a tape archive, lists the files as they are archived, and specifies the tape device.
`v`	Provides additional information about the tar file entries.
`-` (Hyphen)	Represents a placeholder for the tape device.
filenames	Identifies the files to be copied. Separate multiple files with spaces.
`ssh` | *remote-host*	Pipes the `tar` command's output to a remote system.
`dd of=` /dev/rmt/*n*	Represents the output device.
`obs=`*block-size*	Represents the blocking factor.

5. **Remove the tape from the drive. Write the names of the files on the tape label.**

Example 4-10 Copying Files to a Remote Tape Drive (`tar` and `dd`)

```
# tar cvf - * | ssh mercury dd of=/dev/rmt/0 obs=126b
password:
a answers/ 0 tape blocks
a answers/test129 1 tape blocks
a sc.directives/ 0 tape blocks
a sc.directives/sc.190089 1 tape blocks
a tests/ 0 tape blocks
a tests/test131 1 tape blocks
6+9 records in
0+1 records out
```

How to Extract Files From a Remote Tape Device

1. **Insert the tape into the tape drive.**

2. **Change to a temporary directory.**

```
$ cd /var/tmp
```

3. **Extract the files from a remote tape device.**

```
$ ssh remote-host dd if=/dev/rmt/n | tar xvBpf -
```

`ssh` *remote-host*	Indicates a secure shell that is started to extract the files from the tape device by using the `dd` command.

`dd if=/dev/rmt/`*`n`* Indicates the input device.

`| tar xvBpf -` Pipes the output of the `dd` command to the `tar` command, which is used to restore the files.

4. **Verify that the files have been extracted.**

```
$ ls -l
```

Example 4-11 Extracting Files From a Remote Tape Drive

```
$ cd /var/tmp
$ ssh mercury dd if=/dev/rmt/0 | tar xvBpf -
password:
x answers/, 0 bytes, 0 tape blocks
x answers/test129, 48 bytes, 1 tape blocks
20+0 records in
20+0 records out
x sc.directives/, 0 bytes, 0 tape blocks
x sc.directives/sc.190089, 77 bytes, 1 tape blocks
x tests/, 0 bytes, 0 tape blocks
x tests/test131, 84 bytes, 1 tape blocks
$ ls -l
```

Index