



[Oracle Home](#) > [System Administration Guide, Volume 1](#)

> Chapter 36 Mounting and Unmounting File Systems (Tasks)

## System Administration Guide, Volume 1

- [Previous: Chapter 35 Creating File Systems \(Tasks\)](#)
- [Next: Chapter 37 The Cache File System \(Tasks\)](#)

# Chapter 36 Mounting and Unmounting File Systems (Tasks)

This chapter describes how to mount and unmount file systems. This is a list of the step-by-step instructions in this chapter.

- ["How to Determine Which File Systems Are Mounted"](#)
- ["How to Add an Entry to the /etc/vfstab File"](#)
- ["How to Mount a File System \(/etc/vfstab File\)"](#)
- ["How to Mount All File Systems \(/etc/vfstab File\)"](#)
- ["How to Mount a UFS File System"](#)
- ["How to Mount an NFS File System"](#)
- ["IA: How to Mount a System V \(S5FS\) File System"](#)
- ["IA: How to Mount a PCFS \(DOS\) File System From a Hard Disk"](#)
- ["How to Stop All Processes Accessing a File System"](#)
- ["How to Unmount a File System"](#)
- ["How to Unmount All File Systems \(/etc/vfstab File\)"](#)

## Mounting File Systems

After you create a file system, you need to make it available to the system so you can use it. You make a 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. Any other file system can be connected or disconnected from the root (/) file system.

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

Table 36-1 Determining How to Mount File Systems

If You Need to Mount ...	Then You Should Use ...
--------------------------	-------------------------

If You Need to Mount ...	Then You Should Use ...
Local or remote file systems infrequently	The <code>mount</code> command entered manually from the command line.
Local file systems frequently	The <code>/etc/vfstab</code> file, which will mount the file system automatically when the system is booted in multi-user state.
Remote file systems frequently, such as home directories	<ul style="list-style-type: none"> <li>The <code>/etc/vfstab</code> file, which will automatically mount the file system when the system is booted in multi-user state.</li> <li>AutoFS, which will automatically mount or unmount the file system when you change into (mount) or out of (unmount) the directory.</li> </ul> <p>To enhance performance, you can also cache the remote file systems by using CacheFS.</p>

You can mount a CD-ROM containing a file system by simply inserting it into the drive (Volume Management will automatically mount it). You can mount a diskette containing a file system by inserting it into the drive and running the [volcheck\(1\)](#) command. See [Chapter 14, Guidelines for Using CDs and Diskettes \(Overview\)](#) for more information.

## Commands Used to Mount and Unmount File Systems

The table below lists the commands in the `/usr/sbin` directory that you use to mount and unmount file systems.

Table 36-2 Commands for Mounting and Unmounting File Systems

Command	Description
<a href="#">mount(1M)</a>	Mounts file systems and remote resources.
<a href="#">mountall(1M)</a>	Mounts all file systems specified in the <code>/etc/vfstab</code> file. The <code>mountall</code> command is run automatically when entering multiuser run states.

Command	Description
<a href="#">umount(1M)</a>	Unmounts file systems and remote resources.
<a href="#">umountall(1M)</a>	Unmounts all file systems specified in the /etc/vfstab file.

The mount commands will not mount a read/write file system that has known inconsistencies. If you receive an error message from the `mount` or `mountall` command, you might need to check the file system. See [Chapter 39, Checking File System Integrity](#) for information on how to check the file system.

The `umount` commands will not unmount a file system that is busy. A file system is considered busy if a user is accessing a file or directory in the file system, if a program has a file open in that file system, or if the file system is shared.

## Commonly Used Mount Options

The table below describes the commonly used mount options that you can specify with the `-o` option of the `mount` command. If you specify multiple options, separate them with commas (no spaces). For example, `-o ro,nosuid`.

For a complete list of mount options for each file system type, refer to the specific mount command man pages (for example, [mount\\_ufs\(1M\)](#)).

Table 36-3 Commonly Used `-o` Mount Options

Option	File System	Description
<code>bg</code>   <code>fg</code>	NFS	If the first attempt fails, retries in the background ( <code>bg</code> ) or in the foreground ( <code>fg</code> ). This option is safe for non-critical <code>vfstab</code> entries. The default is <code>fg</code> .
<code>hard</code>   <code>soft</code>	NFS	Specifies the procedure if the server does not respond. <code>soft</code> indicates that an error is returned. <code>hard</code> indicates that the retry request is continued until the server responds. The default is <code>hard</code> .
<code>intr</code>   <code>nointr</code>	NFS	Specifies whether keyboard interrupts are delivered to a process that is hung while waiting for a response on a hard-mounted file system. The default is <code>intr</code> (interrupts allowed).

Option	File System	Description
largefiles   nolargefiles	UFS	Enables you to create files larger than 2 Gbytes. The <code>largefiles</code> option means that a file system mounted with this option <b>might</b> contain files larger than 2 Gbytes, but it is not a requirement. The default is <code>largefiles</code> . If the <code>nolargefiles</code> option is specified, the file system could not be mounted on a system running Solaris 2.6 or compatible versions.
logging   nologging	UFS	<p>Enables logging for the file system. UFS logging is the process of storing transactions (changes that make up a complete UFS operation) into a log before the transactions are applied to the UFS file system. Logging helps prevent UFS file systems from becoming inconsistent, which means <code>fsck</code> can be bypassed. Bypassing <code>fsck</code> reduces the time to reboot a system if it crashes, or after a system is shutdown uncleanly.</p> <p>The log is allocated from free blocks on the file system, and is sized approximately 1 Mbyte per 1 Gbyte of file system, up to a maximum of 64 Mbytes. The default is <code>nologging</code>.</p>
noatime	UFS	Suppresses access time updates on files, except when they coincide with updates to the <code>ctime</code> or <code>mtime</code> . See <a href="#">stat(2)</a> . This option reduces disk activity on file systems where access times are unimportant (for example, a Usenet news spool). The default is normal access time ( <code>atime</code> ) recording.
remount	All	Changes the mount options associated with an already-mounted file system. This option can generally be used with any option except <code>ro</code> , but what can be changed with this option is dependent on the file system type.
retry= <i>n</i>	NFS	Retries the mount operation when it fails. <i>n</i> is the number of times to retry.
<code>ro</code>   <code>rw</code>	CacheFS, NFS, PCFS, UFS, S5FS	Specifies read/write or read-only. If you do not specify this option, the default is read/write. The default option for HSFS is <code>ro</code> .

Option	File System	Description
suid   nosuid	CacheFS, HSFS, NFS, S5FS, UFS	Allows or disallows setuid execution. The default is to allow setuid execution.

## How to Determine Which File Systems Are Mounted

You can determine which file systems are mounted by using the `mount` command.

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

<code>-v</code>	Displays the list of mounted file systems in verbose mode.
-----------------	--

### Example--Determining Which File Systems Are Mounted

```
$ mount
/ on /dev/dsk/c0t0d0s0 read/write/setuid/intr/largefiles/onerror=panic on ...
/usr on /dev/dsk/c0t0d0s6 read/write/setuid/intr/largefiles/onerror=panic on ...
/proc on /proc read/write/setuid on Fri Sep 10 16:09:48 1999
/dev/fd on fd read/write/setuid on Fri Sep 10 16:09:51 1999
/etc/mnttab on mnttab read/write/setuid on Fri Sep 10 16:10:06 1999
/var/run on swap read/write/setuid on Fri Sep 10 16:10:06 1999
/tmp on swap read/write/setuid on Fri Sep 10 16:10:09 1999
/export/home on /dev/dsk/c0t0d0s7 read/write/setuid/intr/largefiles/onerror=panic ...
$
```

## Mounting File Systems (`/etc/vfstab` File)

### The `/etc/vfstab` Field Descriptions

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

Table 36-4 `/etc/vfstab` Field Descriptions

Field Name	Description
device to mount	<ul style="list-style-type: none"> <li>The block device name for a local UFS file system (for example, <code>/dev/dsk/c0t0d0s0</code>).</li> <li>The resource name for a remote file system (for example, <code>myserver:/export/home</code>). For more information about NFS, see <a href="#">System Administration Guide, Volume 3</a>.</li> <li>The block device name of the slice on which to swap (for example, <code>/dev/dsk/c0t3d0s1</code>).</li> <li>The <code>/proc</code> directory for the proc file system type.</li> </ul>
device to fsck	<p>The raw (character) device name that corresponds to the UFS file system identified by the <code>device to mount</code> field (for example, <code>/dev/rdsk/c0t0d0s0</code>). This determines the raw interface that is used by <code>fsck</code>. Use a dash (-) when there is no applicable device, such as for a read-only file system or a remote file system.</p>
mount point	<p>Identifies where to mount the file system (for example, <code>/usr</code>).</p>
FS type	<p>The type of file system identified by the <code>device to mount</code> field.</p>
fsck pass	<p>The pass number used by <code>fsck</code> to decide whether to check a file system. When the field contains a dash (-), the file system is not checked.</p> <p>When the field contains a zero, UFS file systems are not checked but non-UFS file systems are checked. When the field contains a value greater than zero, the file system is always checked.</p> <p>All file systems with a value of 1 in this field, are checked one at a time in the order they appear in the <code>vfstab</code> file. When <code>fsck</code> is run on multiple UFS file systems that have <code>fsck</code> pass values greater than one and the preen option (<code>-o p</code>) is used, <code>fsck</code> 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.</p> <p>The <code>fsck</code> pass field does not explicitly specify the order in which file systems are checked, other than as described above.</p>

Field Name	Description
mount at boot	Set to yes or no for whether the file system should be automatically mounted by <code>mountall</code> when the system is booted. Note that this field has nothing to do with AutoFS. The root (/), /usr and /var file systems are not mounted from the <code>fstab</code> file initially. This field should always be set to no for these file systems and for virtual file systems such as /proc and /dev/fd.
mount options	A list of comma-separated options (with no spaces) that are used in mounting the file system. Use a dash (-) to indicate no options. See <a href="#">Table 36-3</a> for a list of commonly used mount options.

**Note -**

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

## How to Add an Entry to the `/etc/vfstab` File

1. Become superuser.

Also, 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.

2. Edit the `/etc/vfstab` file and add an entry.

**Note -**

Since 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.

See [Table 36-4](#) for detailed information about the `/etc/vfstab` field entries. Make sure that you:

- Separate each field with white space (a space or a tab).
- Enter a dash (-) if a field has no contents.

3. Save the changes.

## Examples--Adding an Entry to the `/etc/vfstab` File

The following example mounts the disk slice `/dev/dsk/c0t3d0s7` as a UFS file system attached to the mount point directory `/files1` with the default mount options (read/write). It specifies the raw character device `/dev/rdsks/c0t3d0s7` as the device to fsck. The fsck pass value of 2 means that the file system will be checked, but not sequentially.

#device #to mount #	device to fsck	mount point	FS type	fsck pass	mount at boot	mount options
<code>/dev/dsk/c0t3d0s7</code>	<code>/dev/rdsks/c0t3d0s7</code>	<code>/files1</code>	ufs	2	yes	-

The following example mounts the directory `/export/man` from the system pluto as an NFS file system on mount point `/usr/man`. It does not specify a device to fsck or a fsck pass because it's an NFS file system. In this example, mount options are `ro` (read-only) and `soft`. For greater reliability, specify the `hard` mount option for read/write NFS file systems.

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

The following example mounts the root (/) file system on a loopback mount point named `/tmp/newroot`. It specifies yes for `mount at boot`, no device to fsck, and no `fsck pass` number. LOFS file systems must always be mounted after the file systems used to make up the LOFS file system.

#device #to mount #	device to fsck -	mount point <code>/tmp/newroot</code>	FS type lofs	fsck pass -	mount at boot yes	mount options -
---------------------------	------------------------	---	--------------------	-------------------	-------------------------	-----------------------

## How to Mount a File System (`/etc/vfstab` File)

1. Become superuser.

Also, 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.

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

# <code>mount</code> <i>mount-point</i>	
---	--

<i>mount-point</i>	Specifies an entry in the <code>mount point</code> or <code>device to mount</code> field in the <code>/etc/vfstab</code> file. It is usually easier to specify the mount point.
--------------------	---

## Example--Mounting a File System (/etc/vfstab File)

The following example mounts the /usr/dist file system listed in the /etc/vfstab file.

```
# mount /usr/dist
```

## How to Mount All File Systems (/etc/vfstab File)

1. Become superuser.

Also, 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.

2. Mount the file systems listed in the /etc/vfstab file.

```
# mountall [-l | -r] [-F fstype]
```

If no options are specified, all file systems listed in the /etc/vfstab file with yes in the `mount at boot` field are mounted.

<b>-l</b>	Mounts all the local file systems listed in the /etc/vfstab file with yes in the <code>mount at boot</code> field.
<b>-r</b>	Mounts all the remote file systems listed in the /etc/vfstab file with yes in the <code>mount at boot</code> field.
<b>-F</b> <i>fstype</i>	Mounts all file systems of the specified type listed in the /etc/vfstab file with yes in the <code>mount at boot</code> field.

All the file systems with a `device to fsck` entry are checked and fixed, if necessary, before mounting.

## Examples--Mounting All File Systems (/etc/vfstab File)

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

```
# mountall
/dev/rdsk/c0t0d0s7 already mounted
mount: /tmp already mounted
```

[Cookie Preferences](#) | [Ad Choices](#)

```
mount: /dev/dsk/c0t0d0s7 is already mounted, /export/home is busy,
or the allowable number of mount points has been exceeded
```

The following example mounts all the local systems listed in the /etc/vfstab file.

```
# mountall -l
# mount
/ on /dev/dsk/c0t0d0s0 read/write/setuid/intr/largefiles/onerror=panic on ...
/usr on /dev/dsk/c0t0d0s6 read/write/setuid/intr/largefiles/onerror=panic on ...
/proc on /proc read/write/setuid on Fri Sep 10 16:09:48 1999
/dev/fd on fd read/write/setuid on Fri Sep 10 16:09:51 1999
/etc/mnttab on mnttab read/write/setuid on Fri Sep 10 16:10:06 1999
/var/run on swap read/write/setuid on Fri Sep 10 16:10:06 1999
/tmp on swap read/write/setuid on Fri Sep 10 16:10:09 1999
/export/home on /dev/dsk/c0t0d0s7 read/write/setuid/intr/largefiles/onerror=panic on ...
```

The following example mounts all the remote file systems listed in the /etc/vfstab file.

```
# mountall -r
# mount
/ on /dev/dsk/c0t0d0s0 read/write/setuid/intr/largefiles/onerror=panic on ...
/usr on /dev/dsk/c0t0d0s6 read/write/setuid/intr/largefiles/onerror=panic on ...
/proc on /proc read/write/setuid on Fri Sep 10 16:09:48 1999
/dev/fd on fd read/write/setuid on Fri Sep 10 16:09:51 1999
/etc/mnttab on mnttab read/write/setuid on Fri Sep 10 16:10:06 1999
/var/run on swap read/write/setuid on Fri Sep 10 16:10:06 1999
/tmp on swap read/write/setuid on Fri Sep 10 16:10:09 1999
/export/home on /dev/dsk/c0t0d0s7 read/write/setuid/intr/largefiles/onerror=panic on ...
/usr/dist on mars:/usr/dist remote/read/write/setuid on Tue Sep 14 15:32:18 1999
```

## Mounting File Systems (mount Command)

### How to Mount a UFS File System

1. Become superuser.

Also, 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.

2. Mount the UFS file system by using the `mount` command.

```
# mount [-o mount-options] /dev/dsk/device-name mount-point
```

**-o mount-options**

Specifies mount options that you can use to mount a UFS file system. See [Table 36-3](#) or [mount\\_ufs\(1M\)](#) for a list of options.

<i>/dev/dsk/device-name</i>	Specifies the disk device name for the slice holding the file system (for example, <i>/dev/dsk/c0t3d0s7</i> ). See " <a href="#">How to Display Disk Slice Information</a> " to get slice information for a disk.
<i>mount-point</i>	Specifies the directory on which to mount the file system.

## Example--Mounting a UFS File System

The following example mounts */dev/dsk/c0t3d0s7* on the */files1* directory.

```
# mount /dev/dsk/c0t3d0s7 /files1
```

## Example--Mounting a UFS File System With Logging Enabled

UFS logging eliminates file system inconsistency, which can significantly reduce the time of system reboots. The following example mounts */dev/dsk/c0t3d0s7* on the */files1* directory with logging enabled.

```
# mount -o logging /dev/dsk/c0t3d0s7 /files1
```

## How to Remount a UFS File System Without Large Files

When you mount a file system, the *largefiles* option is selected by default, which enables you to create files larger than 2 Gbytes. Once a file system contains large files, you cannot remount the file system with the *nolargefiles* option or mount it on a system running Solaris 2.6 or compatible versions, until you remove any large files and run *fsck* to reset the state to *nolargefiles*.

This procedure assumes that the file system is in the */etc/vfstab* file.

1. Become superuser.
2. Make sure there are no large files in the file system.

```
# cd mount-point
# find . -xdev -size +20000000 -exec ls -l {} \;
```

<i>mount-point</i>	Specifies the mount point of the file system you want to check for large files.
--------------------	---

3. Unmount the file system.

```
# umount mount-point
```

4. Reset the file system state.

```
# fsck mount-point
```

5. Remount the file system with the `nolargefiles` option.

```
# mount -o nolargefiles mount-point
```

## Example--Mounting a File System Without Large Files

The following example checks the `/datab` file system and remounts it with the `nolargefiles` option.

```
# cd /datab
# find . -xdev -size +20000000 -exec ls -l {} \;
# umount /datab
# fsck /datab
# mount -o nolargefiles /datab
```

## How to Mount an NFS File System

1. Become superuser.

Also, 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.

2. Make sure 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. See [System Administration Guide, Volume 3](#) for information on how to share resources.

3. Mount the NFS file system by using the `mount` command.

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

**-o *mount-options***

Specifies `mount` options that you can use to mount an NFS file system. See [Table 36-3](#) for the list of commonly used `mount` options or [mount nfs\(1M\)](#) for a

[Cookie Preferences](#) | [Ad Choices](#)

	complete list of options.
<i>server:/directory</i>	Specifies the server's host name that contains the shared resource, and the path to the file or directory to mount.
<i>mount-point</i>	Specifies the directory on which to mount the file system.

## Example--Mounting an NFS File System

The following example mounts the `/export/packages` directory on `/mnt` from the server pluto.

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

## IA: How to Mount a System V (S5FS) File System

1. Become superuser.

Also, 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.

2. Mount the S5FS file system by using the `mount` command.

```
# mount -F s5fs [-o mount-options] /dev/dsk/device_name mount-point
```

<i>-o mount-options</i>	Specifies mount options that you can use to mount a S5FS file system. See <a href="#">Table 36-3</a> for the list of commonly used mount options or <a href="#">mount_s5fs(1M)</a> for a complete list of options.
<i>/dev/dsk/device-name</i>	Specifies the disk device name of the slice holding the file system (for example, <code>/dev/dsk/c0t3d0s7</code> ). See <a href="#">"How to Display Disk Slice Information"</a> to get slice information for a disk.
<i>mount-point</i>	Specifies the directory on which to mount the file system.

## IA: Example--Mounting an S5FS File System

The following example mounts /dev/dsk/c0t3d0s7 on the /files1 directory.

```
# mount -F s5fs /dev/dsk/c0t3d0s7 /files1
```

## IA: How to Mount a PCFS (DOS) File System From a Hard Disk

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

1. Become superuser.

Also, 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.

2. Mount the PCFS file system by using the `mount` command.

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

<b>-o rw   ro</b>	Specifies that you can mount a PCFS file system read/write or read-only. If you do not specify this option, the default is read/write.
<i>/dev/dsk/device-name</i>	Specifies the device name of the whole disk (for example, /dev/dsk/c0t0d0p0).
<i>logical-drive</i>	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.
<i>mount-point</i>	Specifies the directory on which to mount the file system.

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

## IA: Examples--Mounting a PCFS (DOS) File System From a Hard Disk

The following example mounts 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 mounts the first logical drive in the Extended DOS slice read-only on /mnt.

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

## Unmounting File Systems

Unmounting a file system removes it from the file system mount point, and deletes the entry from the /etc/mnttab file. Some file system administration tasks cannot be performed on mounted file systems. You should unmount a file system when:

- It is no longer needed or has been replaced by a file system that contains more current software.
- You need to check and repair it using the fsck command. See [Chapter 39, Checking File System Integrity](#) for more information about the fsck command.

It is a good idea to unmount a file system before doing a complete backup. See [Chapter 43, Backing Up Files and File Systems \(Tasks\)](#) for more information about doing backups.

---

### Note -

File systems are automatically unmounted as part of the system shutdown procedure.

---

## Prerequisites

The prerequisites to unmounting file systems are:

- You must be superuser.
  - A file system must be available for unmounting. 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 it is being shared. You can make a file system available for unmounting by:
    - Changing to a directory in a different file system.
    - Logging out of the system.
    - Using the fuser command to list all processes accessing the file system and to stop them if necessary. See ["How to Stop All Processes Accessing a File System"](#) for more details.
- Notify users if you need to unmount a file system they are using.
- Unsharing the file system

## Verifying an Unmounted File System

To verify that you unmounted a file system or a number of file systems, look at the output from the mount command. This is described in ["How to Determine Which File Systems Are Mounted"](#)

[Cookie Preferences](#) | [Ad Choices](#)

## How to Stop All Processes Accessing a File System

1. Become superuser.
2. List all the processes that are accessing the file system, so you know which processes you are going to stop.

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

<b>-c</b>	Reports on files that are mount points for file systems and any files within those mounted file systems.
<b>-u</b>	Displays the user login name for each process ID.
<i>mount-point</i>	The name of the file system for which you want to stop processes.

3. Stop all processes accessing the file system.

---

**Note -**

You should not stop a user's processes without warning.

---

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

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

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

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

### Example--Stopping All Processes Accessing a File System

The following example stops process 4006c that is using the /export/home file system.

```
# fuser -c /export/home
```

```
# 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 (except /, /usr, or /var):

### Note -

The root (/), /usr, and /var file systems are special cases. The root (/) file system can be unmounted only during a shutdown, since the system needs the root (/) file system to function.

1. Make sure you have met the prerequisites listed on "["Prerequisites"](#)".
2. Unmount the file system.

<pre># umount <i>mount-point</i></pre>	<p><i>mount-point</i>      The name of the file system that you want to unmount. This can either be the directory name where the file system is mounted, the device name path of the file system, the resource for an NFS file system, or the loopback directory for LOFS file systems.</p>
--	---

## Examples--Unmounting a File System

The following example unmounts a local home file system.

<pre># umount /export/home</pre>
----------------------------------

The following example unmounts the file system on slice 7.

<pre># umount /dev/dsk/c0t0d0s7</pre>
---------------------------------------

## How to Unmount All File Systems (*/etc/vfstab* File)

Use the following procedure to unmount all the file systems listed in the /etc/vfstab file, except for the /, /proc, /var, and /usr file systems.

1. Make sure you have met the prerequisites listed on "["Prerequisites"](#)".

2. Unmount all the file systems listed in the `/etc/vfstab` file.

```
# umountall
```

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

3. For the file systems that were busy and not unmounted, make them available to be unmounted as described in ["How to Stop All Processes Accessing a File System"](#).

4. Repeat Step 2 as needed until all file systems are unmounted.

- [Previous: Chapter 35 Creating File Systems \(Tasks\)](#)
- [Next: Chapter 37 The Cache File System \(Tasks\)](#)
- © 2010, Oracle Corporation and/or its affiliates