VERITAS

VERITAS® File System

Quick Start Guide

Release 3.3.2 - Binary

Solaris

May 1999

P/N 100-001048

TRADEMARKS

VERITAS File System Quick Start Guide

Introduction

This guide provides instructions and examples on performing the following VERITAS File System™ (VxFS® or `vxfs`) operations:

- Creating a File System
- Mounting a File System
- Unmounting a File System
- Displaying Mounted File Systems
- Identifying File System Types
- Resizing a File System
- Backing Up and Restoring a File System
- Using Quotas

Audience

This guide is for system administrators who configure and maintain operating systems that use VERITAS File Systems. This guide assumes you have:

- An understanding of system administration.
- A working knowledge of the UNIX operating system.
- A general understanding of file systems.

Related Documents

- The *VERITAS File System System Administrator's Guide* describes VERITAS File System concepts, how to use various utilities, and how to perform backup procedures.
- The *VERITAS File System Installation Guide* provides instructions on how to install VxFS and verify the installation.

Getting Help

For license information or information about VERITAS service packages, contact VERITAS Customer Support.

U.S. and Canadian Customers: 1-800-342-0652

International Customers: +1 (650) 335-8555

FAX: (650) 335-8428

Electronic mail: `support@veritas.com`

For additional information about VERITAS and VERITAS products, visit the website at:

```
www.veritas.com
```

Conventions

The following table describes the typographic conventions used in this guide.

Typeface	Usage	Examples

`courier`	Computer output, user input, command names, files, and directories	`$ You have mail.`
		The `cat` command displays files.
		`$ ls -a`

italics	New terms, document titles, words to be emphasized, glossary cross references, variables to be substituted with a real name or value	`$ cat` *filename* See the *User's Guide* for details.
Symbol	**Usage**	
%	C shell prompt	
$	Bourne/Korn shell prompt	
#	Superuser prompt (all shells)	

\	In examples showing user input, indicates continued input on the following line; you do not type this character	`# mount -F vxfs \`
		`/h/filesys`

[ ]$	In a command synopsis, brackets indicates an optional argument.	`# ls [-a]`
|$	In a command synopsis, a vertical bar separates mutually exclusive arguments.	`# mount [suid|nosuid]`

Creating a File System

The mkfs command creates a VxFS file system by writing to a special character device file. The special character device is a raw disk device or a VERITAS Volume Manager™ (VxVM®) volume. mkfs builds a file system with a root directory and a lost+found directory. See the fsck(1M) manual page for more information.

Before running mkfs, you must initialize the target device:

- If you have added a new kind of disk controller, which requires a new driver, you must run drvconfig.
- If you have added a new disk, you must run disks, then run format to create a disk slice.

  The drvconfig(1M), disks(1M), and format(1M)) manual pages explain how to perform these tasks. If you are using a logical device (such as a VERITAS Volume Manager volume), see the appropriate VxVM documentation for instructions on device initialization.

How to Create a File System

To create a file system, use the mkfs command:

```
mkfs [-F vxfs] [generic_options] [-o specific_options] special [size]
```

vxfs is the file system type, *generic_options* are the options common to most other file system types, *specific_options* are options specific to the VERITAS file system, *special* is the character (raw) device or VERITAS Volume Manager volume, and *size* is the size of the new file system (in sectors).

If you specify the -o N flag, mkfs only displays the geometry of the file system and does not write to the device.

If you specify the -o largefiles flag, mkfs can create files larger than two gigabytes.

For more information about the mkfs command and its available options, see:

- mkfs(1M)
- mkfs_vxfs(1M)

Example

To create a VxFS file system 12288 sectors in size on `/dev/rdsk/c0t6d0s2`, enter:

```
# mkfs -F vxfs /dev/rdsk/c0t6d0s2 12288
```

Information similar to the following displays:

```
version 2 layout

12288 sectors, 6144 blocks of size 1024, log size 512 blocks
unlimited inodes, 5597 data blocks, 5492 free data blocks
1 allocation units of 32778 blocks, 32768 data blocks
last allocation unit has 5597 data blocks
first allocation unit starts at block 537
overhead per allocation unit is 10 blocks
initial allocation overhead is 105 blocks

version 4 layout

12288 sectors, 6144 blocks of size 1024, log size 512 blocks
unlimited inodes, 5597 data blocks, 5492 free data blocks
1 allocation units of 32778 blocks, 32768 data blocks
last allocation unit has 5597 data blocks
first allocation unit starts at block 537
overhead per allocation unit is 10 blocks
initial allocation overhead is 105 blocks
```

At this point you can mount the newly created file system.

Mounting a File System

You mount a VERITAS file system by using the `mount` command. When you enter the `mount` command, the generic `mount` command parses the arguments. If the `-F` *FSType* option is specified, the `mount` command specific to that file system type executes. It first searches the directory `/etc/fs/`*FSType*, then the `/usr/lib/fs/`*FSType* directory. If the `-F` option is not supplied, the command searches the file `/etc/vfstab` for a file system and *FSType* matching the special file or mount point provided. If no file system type is specified, `mount` uses the default file system.

How to Mount a File System

After creating a file system, you mount it with the `mount` command:

```
mount [-F vxfs] [generic_options] [-r] [-o specific_options] \
special mount_point
```

`vxfs` is the file system type, *generic_options* are the options common to most other file system types, *specific_options* are options specific to `vxfs`, *special* is a block special device, and *mount_point* is the directory on which to mount the file system. The `-r` option mounts the file system as read-only.

Mount Options

The `mount` command has numerous options to tailor it for various functions and environments. Some *specific_options* are listed below.

- Security feature

  If security is important, use `blkclear` to ensure that deleted files are completely erased before the space is reused.

- Support for large files

  If you specify the `largefiles` option, then the file system can contain files greater than 2 gigabytes.

- Using databases

  If you are using databases with VxFS and if you have installed a license key for the VERITAS Quick I/O™ for Databases feature, the `mount` command enables Quick I/O by default. The `noqio` option disables Quick I/O. If you do not have Quick I/O, `mount` ignores the `qio` option. Alternatively, you can increase database performance using the mount option `convosync=direct`, which utilizes direct I/O.

- News file systems

  If you are using `cnews`, use `delaylog` (or `tmplog`)`,mincache= closesync` because `cnews` does an `fsync()` on each news file before marking it received. The `fsync()` is performed synchronously as required, but other options are delayed.

- VERITAS QuickLog™

  If you are using QuickLog, you can improve I/O performance by moving logging to a separate disk device by using `vxldlog=`[*dev*]. See the "VERITAS QuickLog" chapter of the *VERITAS File System System Administrator's Guide* for more information.

- Temporary file systems

  For a temporary file system such as `/tmp`, where performance is more important than data integrity, use `tmplog,mincache=tmpcache`.

For more information about the `mount` command and its available options, see:

- `mount`(1M)
- `mount_vxfs`(1M)
- `vfstab`(4)
- The *VERITAS File System System Administrator's Guide*

Example

To mount the file system /dev/dsk/c0t6d0s2 on the /ext directory with read/write access and delayed logging, enter:

```
# mount -F vxfs -o delaylog /dev/dsk/c0t6d0s2 /ext
```

How to Edit the vfstab File

You can edit the /etc/vfstab file to automatically mount a file system at boot time. You must specify:

- the special block device name to mount
- the special character device name used by fsck
- the mount point
- the mount options
- the file system type (vxfs)
- which fsck pass looks at the file system
- whether to mount the file system at boot time

Each entry must be on a single line. The file format is described in the vfstab(4) manual page.

Here is a typical vfstab file with the new file system on the last line:

```
#device          device              mount     FS    fsck   mount     mount
#to mount        to fsck             point     type  pass   at boot  options
#
#/dev/dsk/c1d0s2/dev/rdsk/c1d0s2     /usr      ufs   1      yes       -
/proc            -                   /proc     proc  -      no        -
fd               -                   /dev/fd   fd    -      no        -
swap             -                   /tmp      tmpfs -      yes       -
/dev/dsk/c0t3d0s0/dev/rdsk/c0t3d0s0  /         ufs   1      no        -
/dev/dsk/c0t3d0s1-                   -         swap  -      no        -
/dev/dsk/c0t6d0s2/dev/rdsk/c0t6d0s2  /ext      vxfs  1      yes       -
```

Unmounting a File System

Use the `umount` command to unmount a currently mounted file system.

How to Unmount a File System

To unmount a file system, use the following syntax:

```
umount special | mount_point
```

Specify the file system to be unmounted as a *mount_point* or *special* (the device on which the file system resides).

For more information about the `umount` command and its available options, see:

- `umount`(1M)

Example

To unmount the file system `/dev/dsk/c0t6d0s2`, enter:

```
# umount /dev/dsk/c0t6d0s2
```

To unmount all file systems not required by the system, enter:

```
# umount -a
```

This unmounts all file systems except `/`, `/usr`, `/usr/kvm`, `/var`, `/proc`, `/dev/fd`, and `/tmp`.

Displaying Mounted File Systems

You can use the `mount` command to display a list of currently mounted file systems. It is important to keep track of which file systems are mounted and which are not. This saves users from trying to access unmounted file systems.

How to Display a File System

To see the status of mounted file systems, use the syntax:

```
mount -v
```

This shows the file system type and `mount` options for all mounted file systems. The `-v` option specifies verbose mode.

For more information about the `mount` command and its available options, see:

- `mount`(1M)
- `mount_vxfs`(1M)

Example

When invoked without options, the `mount` command displays file system information similar to the following:

```
# mount

/ on /dev/root read/write/setuid on Thu May 26 16:58:24 1996
/proc on /proc read/write on Thu May 26 16:58:25 1996
/dev/fd on /dev/fd read/write on Thu May 26 16:58:26 1996
/tmp on /tmp read/write on Thu May 26 16:59:33 1996
/var/tmp on /var/tmp read/write on Thu May 26 16:59:34 1996
```

Identifying File System Types

Use the `fstyp` command to determine the file system type for a specified file system. This is useful when someone else has created a file system and you want to know what kind it is.

How to Identify a File System

To determine the status of mounted file systems, use the syntax:

```
fstyp -v special
```

special is the character (or raw) device. The `-v` option specifies verbose mode.

For more information about the `fstyp` command and its available options, see:

- `fstyp`(1M)
- `fstyp_vxfs`(1M)

Example

To find out what kind of file system is on the device `/dev/rdsk/c0t6d0s2`, enter:

```
# fstyp -v /dev/rdsk/c0t6d0s2
```

The output indicates that the file system type is `vxfs`, and displays file system information similar to the following:

```
vxfs
magic a501fcf5  version 2  ctime Tue Oct 23 18:29:39 1998
logstart 17  logend 1040
bsize  1024 size  1048576 dsize  1047255  ninode 0  nau 8
defiextsize 64  ilbsize 0  immedlen 96  ndaddr 10
aufirst 1049  emap 2  imap 0  iextop 0  istart 0
bstart 34  femap 1051  fimap 0  fiextop 0  fistart 0  fbstart 1083
nindir 2048  aulen 131106  auimlen 0  auemlen 32
auilen 0  aupad 0  aublocks 131072  maxtier 17
inopb 4  inopau 0  ndiripau 0  iaddrlen 8   bshift 10
inoshift 2  bmask fffffc00  boffmask 3ff  checksum d7938aa1
oltext1 9  oltext2 1041  oltsize 8  checksum2 52a
free 382614  ifree 0
efree  676 413 426 466 612 462 226 112 85 35 14 3 6 5 4 4 0 0
#
```

Resizing a File System

You can extend or shrink mounted VERITAS file systems using the `fsadm` command.

See the following manual pages for more information on resizing file systems:

- `format`(1M)
- `fsadm_vxfs`(1M)

How to Extend a File System Using `fsadm`

If a VERITAS file system is not large enough, you can increase its size. The size of the file system is specified in units of 512-byte blocks (or sectors).

To extend a VERITAS file system, use the syntax:

```
/usr/lib/fs/vxfs/fsadm [-b newsize] [-r rawdev] mount_point
```

`vxfs` is the file system type, *newsize* is the size (in sectors) to which the file system will increase, and *mount_point* is the file system's mount point.

You can use the `-r` *rawdev* option to specify the pathname of the raw device if there is no entry in `/etc/vfstab` and `fsadm` cannot determine the raw device.

Note: The device must be big enough to contain the new larger file system. See the `format`(1M) manual page or the VERITAS *Volume Manager System Administrator's Guide* for more information.

Example

To extend the VERITAS file system mounted at `/ext` to 22528 sectors, enter:

```
# /usr/lib/fs/vxfs/fsadm -b 22528 /ext
```

How to Shrink a File System

You can decrease the size of the file system using `fsadm`, even while the file system is mounted.

Note: In cases where data is allocated towards the end of the file system, shrinking may not be possible.

To decrease the size of a VERITAS file system, use the syntax:

```
/usr/lib/fs/vxfs/fsadm [-b newsize] [-r rawdev] mount_point
```

`vxfs` is the file system type, *newsize* is the size (in sectors) to which the file system will shrink, and *mount_point* is the file system's mount point.

You can use the `-r` *rawdev* option to specify the pathname of the raw device if there is no entry in `/etc/vfstab` and `fsadm` cannot determine the raw device.

Example

To shrink a VERITAS file system mounted at `/ext` to 20480 sectors, enter:

```
# /usr/lib/fs/vxfs/fsadm -b 20480 /ext
```

Note: After this operation, there is unused space at the end of the device. You can now resize the device. Be careful not to make the device smaller than the new size of the file system.

How to Reorganize a File System

You can reorganize (or compact) a fragmented file system using fsadm, even while the file system is mounted. This may help shrink a file system that could not previously be decreased.

To reorganize a VERITAS file system, use the syntax:

```
/usr/lib/fs/vxfs/fsadm [-e] [-d] [-E] [-D] \
[-r rawdev] mount_point
```

vxfs is the file system type, -e reorganizes extents, -d reorganizes directories, -E reports on extent fragmentation, -D reports on directory fragmentation, and *mount_point* is the file system's mount point.

You can use the -r *rawdev* option to specify the pathname of the raw device if there is no entry in /etc/vfstab and fsadm cannot determine the raw device.

Example

To reorganize the VERITAS file system mounted at /ext, enter:

```
# /usr/lib/fs/vxfs/fsadm -EeDd /ext
```

Backing Up and Restoring a File System

To back up a VERITAS file system, you first create a read-only snapshot file system, then back up the snapshot. This procedure lets you keep the main file system on line. The snapshot is a copy of the *snapped* file system that is frozen at the moment it is created.

See the following for more information about the mount, vxdump, and vxrestore commands and their available options:

- mount(1M)
- mount_vxfs(1M)
- vxdump(1M)
- vxrestore(1M)
- *VERITAS File System System Administrator's Guide*

How to Create and Mount a Snapshot File System

The first step in backing up a `vxfs` file system is to create and mount a snapshot file system.

To create and mount a snapshot of a VxFS file system, use the syntax:

```
mount [-F vxfs] -o snapof=source,[snapsize=size] \
destination  snap_mount_point
```

source is the special device name or mount point of the file system to copy, *destination* is the name of the special device on which to create the snapshot, *size* is the size of the snapshot file system, and *snap_mount_point* is where to mount the snapshot. *snap_mount_point* must exist before you enter this command.

Example

To create a snapshot file system of the file system at `/dev/dsk/c0t6d0s2` on `/dev/dsk/c0t5d0s2` and mount it at `/snapmount`, enter:

```
# mount -F vxfs -o snapof=/dev/dsk/c0t6d0s2, \
  snapsize=32768 /dev/dsk/c0t5d0s2 /snapmount
```

You can now back up the file system, as described in the following section.

How to Back Up a File System

After creating a snapshot file system as described in the previous section, you can use `vxdump` to back it up.

To back up a VxFS snapshot file system, use the syntax:

```
vxdump [-c] [-f backupdev] snap_mount_point
```

The `-c` option specifies using a cartridge tape device, *backupdev* is the device on which to back up the file system, and *snap_mount_point* is the snapshot file system’s mount point.

Example

To back up the VxFS snapshot file system mounted at `/snapmount` to the tape drive with device name `/dev/rmt/00m`, enter:

```
# vxdump -cf /dev/rmt/00m /snapmount
```

How to Restore a File System

After backing up the file system, you can restore it using the `vxrestore` command. First, create and mount an empty file system.

To restore a VxFS snapshot file system, use the syntax:

```
vxrestore [-v] [-x] mount_point
```

The `-v` option specifies verbose mode. `-x` extracts the named files from the tape. *mount_point* is the restored file system's mount point.

Example

To restore a VxFS snapshot file system using `/restore` as a mount point, enter:

```
# vxrestore -vx /restore
```

Using Quotas

You can use Berkeley Software Distribution (BSD) quotas to allocate per-user quotas on VERITAS file systems.

See the following manual pages for more information on the `vxquota`, `vxquotaon`, `vxquotaoff`, and `vxedquota` commands and their available options:

- `vxquota`(1M)
- `vxquotaon`(1M)
- `vxquotaoff`(1M)
- `vxedquota`(1M)
- *VERITAS File System System Administrator's Guide*

How to Turn On Quotas

You can enable quotas at mount time or after a file system is mounted. The root directory of the file system must contain a file named `quotas` that is owned by `root`.

To turn on quotas for a mounted file system, use the syntax:

```
vxquotaon mount_point
```

To mount a file system and turn on quotas at the same time, use the syntax:

```
mount -F vxfs -o quota special mount_point
```

If there is no `quotas` file in the root directory, the `mount` command succeeds, but `quotas` are not turned on.

Example

To turn on quotas for a VxFS file system mounted at `/mnt` and create the quotas file (if it does not already exist), enter:

```
# touch /mnt/quotas
# vxquotaon /mnt
```

To turn on quotas for a file system at mount time, enter:

```
# mount -F vxfs -o quota /dev/dsk/c0t5d0s2 /mnt
```

How to Set Up User Quotas

You can set user quotas with the `vxedquota` command if you have superuser privileges. User quotas can have a *soft limit* or *hard limit*. You can modify the limits or assign them specific values. Users are allowed to exceed the soft limit, but only for a specified time. Disk usage can never exceed the hard limit. The default limit is seven days on VERITAS file systems.

`vxedquota` creates a temporary file for a specified user. This file contains on-disk quotas for each mounted VxFS file system that has a `quotas` file. The temporary file has one or more lines similar to:

```
fs /mnt blocks (soft = 0, hard = 0) inodes (soft=0, hard=0)
fs /mnt1 blocks (soft = 100, hard = 200) inodes (soft=10, hard=20)
```

Quotas do not need to be turned on for `vxedquota` to work. However, the quota limits apply only after quotas are turned on for a given file system.

`vxedquota` has an option to modify time limits. Modified time limits apply to the entire file system; limits cannot be set for an individual user.

To invoke the quota editor, use the syntax:

`vxedquota` *username*

To modify the time limit, use the syntax:

```
vxedquota -t
```

How to View Quotas

The superuser or individual users can view their disk quotas and usage on VERITAS file systems using the `vxquota` command.

To view quotas for a given user, use the syntax:

```
vxquota -v username
```

This displays the user’s quotas and disk usage on all mounted VERITAS file systems where the `quotas` file exists.

How to Turn Off Quotas

You can turn off quotas for a mounted file system using the `vxquotaoff` command.

To turn off quotas for a file system, use the syntax:

```
vxquotaoff mount_point
```

Example

To turn off quotas for a VxFS file system mounted at `/mnt`, enter:

```
# vxquotaoff /mnt
```