Tru64 UNIX

AdvFS Administration

Part Number: AA-RH96C-TE

June 2001

Product Version: Tru64 UNIX Version 5.1A or higher

This manual provides task-based information and instruction for understanding, configuring, troubleshooting, and tuning the AdvFS file system. It is designed primarily for system administrators.

Compaq Computer Corporation
Houston, Texas

Compaq shall not be liable for technical or editorial errors or omissions contained herein. The information in this publication is subject to change without notice and is provided "as is" without warranty of any kind. The entire risk arising out of the use of this information remains with recipient. In no event shall Compaq be liable for any direct, consequential, incidental, special, punitive, or other damages whatsoever (including without limitation, damages for loss of business profits, business interruption or loss of business information), even if Compaq has been advised of the possibility of such damages.

The limited warranties for Compaq products are exclusively set forth in the documentation accompanying such products. Nothing herein should be construed as constituting a further or additional warranty.

Contents

2 Managing Quotas

3 Backing Up and Restoring Data

4 Optimizing and Tuning the AdvFS File System

5 Troubleshooting

A Accessing File System Information with the SysMan Menu

B AdvFS Commands

C Converting File Systems

D Graphical User Interfaces

Figures

Tables

About This Manual

This manual provides task-based information and instruction for understanding, configuring, troubleshooting, and tuning the Advanced File System (AdvFS) file system. AdvFS is the default file system for the Compaq Tru64™ UNIX operating system. For hardware and software requirements, see the *Tru64 UNIX Software Product Description (SPD)*.

AdvFS Utilities is a separately licensed product that provides additional file management capabilities and graphical user interfaces to simplify system administration. Logical Storage Manager Storage Administrator (LSMSA) provides a java-based graphical interface for managing LSM objects and the AdvFS file system (see *Logical Storage Manager*). The AdvFS GUI, which runs under the Common Desktop Environment (CDE), lets you perform AdvFS operations in a graphical environment.

Audience

This manual is designed primarily for system administrators. It describes the AdvFS file system and AdvFS Utilities. Administrators and programmers who are familiar with other UNIX systems can use this manual as a teaching guide for the AdvFS file system.

New and Changed Features

This manual documents the following features that are new in AdvFS in Version 5.1A:

- A new domain recovery tool, `fixfdmn`, that examines metadata corruption (see *Section 5.8.3*).
- A new mount option for temporary atomic-write data logging for a file (see *Section 4.4*).
- Optional object safety (object reuse) protection. Pages on disk belonging to a fileset can be zero-filled and forced to disk before they are available to the file (see *Section 1.7.10*).
- The ability to turn frag files on and off by using the `mkfset` and `chfsets` commands (see *Section 4.2*).

The following features were added to AdvFS in previous Version 5 releases:

- AdvFS cache tuning is done through the UBC parameters (see *Section 4.5*).

- Improved disk structure that increases the number of files the domain can track (see *Section 1.6.3*)
- A disk salvage utility that can recover information at the block level from disks that have been damaged (see *Section 5.8.4*)
- An improved directory structure that increases the speed of file creation and access (see *Section 1.6.3.2*)
- Enhanced `vdump` and `vrestore` command capability (see *Chapter 3*)
- Remote device support for backup and restore (see *Chapter 3*)
- Increased quota limits (see *Chapter 2*)
- Direct I/O to allow unbuffered, synchronous I/O (see *Section 4.6*)
- Smooth sync option to promote continuous I/O (see *Section 4.7*)
- New utilities (such as `vdf`, which displays disk usage for domains and filesets) (see *Section 2.3.4.4*)
- Metadata display utilities (see *Section 5.13.1*)

You can review all documentation for all releases at **`http://www.tru64unix.compaq.com/docs/`** Also at this site you can view the Technical Updates for any information not included in the documentation provided with your media.

Organization

This manual is organized as follows:

Chapter 1
Describes the volumes, domains, and filesets that make up the AdvFS file system and provides information for configuring the system

Chapter 2
Explains user and group quotas as well as fileset quotas, which are unique to the AdvFS file system

Chapter 3
Provides directions for backing up and restoring data

Chapter 4
Provides guidelines for optimizing and tuning your AdvFS file system

Chapter 5
Provides information to help you troubleshoot problems with your AdvFS file system

Appendix A
Explains the use of the SysMan menu

Appendix B
Lists all the AdvFS commands and compares them with UFS commands

Appendix C
Provides an explanation and sample code for converting a file system to AdvFS

Appendix D	Describes the AdvFS Graphical User Interface
Appendix E	Describes using controller-based cloning and snapshotting

Related Documentation

Other useful documentation includes:

- *System Administration* describes the tasks you perform to administer the operating system running on a workstation or server.
- The *Installation Guide* describes how to install your operating system software.
- *Logical Storage Manager* provides information on storage management.
- The *Programmer's Guide* describes the development environment of the Tru64 UNIX operating system.
- The *Technical Overview* provides a brief overview of Tru64 UNIX and its components.
- *Software License Management* describes how to use the License Management Facility (LMF) to manage software licenses on a Tru64 UNIX system.
- *System Configuration and Tuning* provides information on system performance tuning and advanced kernel configuration.
- *Cluster Administration* describes how to manage systems that run TruCluster software.

  Best Practice documentation illustrates the optimal way to perform common administrative tasks.
- *System Limits*, *Release Notes*, Software Product Descriptions for Tru64 UNIX and AdvFS Utilities, Best Practices, and other documentation are also available.

The *Documentation Overview* provides information on all of the books in the Tru64 UNIX documentation set.

The Tru64 UNIX documentation is available on the World Wide Web at the following URL:

```
http://www.tru64unix.compaq.com/docs/
```

Icons on Tru64 UNIX Printed Books

The printed version of the Tru64 UNIX documentation uses letter icons on the spines of the books to help specific audiences quickly find the books that meet their needs. (You can order the printed documentation from Compaq.) The following list describes this convention:

G	Books for general users
S	Books for system and network administrators
P	Books for programmers
D	Books for device driver writers
R	Books for reference page users

Some books in the documentation help meet the needs of several audiences. For example, the information in some system books is also used by programmers. Keep this in mind when searching for information on specific topics.

`http://www.tru64unix.compaq.com/docs/` provides information on all of the books in the Tru64 UNIX documentation set.

Reader's Comments

Compaq welcomes any comments and suggestions you have on this and other Tru64 UNIX manuals.

You can send your comments in the following ways:

- Fax: 603-884-0120 Attn: UBPG Publications, ZKO3-3/Y32
- Internet electronic mail: `readers_comment@zk3.dec.com`

  A Reader's Comment form is located on your system in the following location:

  `/usr/doc/readers_comment.txt`

Please include the following information along with your comments:

- The full title of the manual and the order number. (The order number appears on the title page of printed and PDF versions of a manual.)
- The section numbers and page numbers of the information on which you are commenting.
- The version of Tru64 UNIX that you are using.
- If known, the type of processor that is running the Tru64 UNIX software.

The Tru64 UNIX Publications group cannot respond to system problems or technical support inquiries. Please address technical questions to your local system vendor or to the appropriate Compaq technical support office. Information provided with the software media explains how to send problem reports to Compaq.

Conventions

This guide uses the following conventions:

#

A number sign represents the superuser prompt.

% **cat**

Boldface type in interactive examples indicates typed user input.

file

Italic (slanted) type indicates variable values, placeholders, and function argument names.

[|] { | }

In syntax definitions, brackets indicate items that are optional and braces indicate items that are required. Vertical bars separating items inside brackets or braces indicate that you choose one item from among those listed.

. . .

In syntax definitions, a horizontal ellipsis indicates that the preceding item can be repeated one or more times.

cat(1)

A cross-reference to a reference page includes the appropriate section number in parentheses. For example, cat(1) indicates that you can find information on the cat command in Section 1 of the reference pages.

Ctrl/*x*

This symbol indicates that you hold down the first named key while pressing the key that follows the slash.

1

Configuring the File System

The Advanced File System (AdvFS) is the default file system for the Compaq Tru64 UNIX operating system. This chapter introduces the file system and explains how to configure and AdvFS root file system:

- Section 1.1 compares AdvFS with UFS.
- Section 1.2 explains license registration for additional AdvFS functionality.
- Section 1.3 explains the design and components of an AdvFS file system.
- Section 1.4 is an overview for setting up the file system.
- Section 1.5 explains volumes, the storage units of AdvFS.
- Section 1.6 examines domains, the physical storage layer of the file system.
- Section 1.7 explains filesets, which form the directory hierarchy of AdvFS.
- Section 1.8 describes how to set up your system to make deleted files retrievable.
- Section 1.9 describes how to configure an AdvFS root file system.
- Section 1.10 introduces the use of LSM with AdvFS.

1.1 What is AdvFS?

The AdvFS file system differs from the traditional UNIX File System (UFS). The UFS model is rigid. Each disk (or disk partition) contains one separate file system; you mount the file system into the logical name space using mount points.

The directory hierarchy layer of UFS is bound tightly to the physical storage layer. When a file system becomes full, this tight binding makes it impossible to move selected files onto another disk without changing the full pathnames of those files. The task of dividing a logical directory into directory subtrees and mapping the subtrees onto separate disks requires careful consideration. Even with extensive planning, adjustments to the directory structure are limited with the UFS model.

In contrast, with AdvFS you can modify your system configuration at any time without shutting down the system. As your system requirements change, AdvFS allows you to easily add or remove storage to meet your requirements.

From a user's perspective, AdvFS looks like any other UNIX file system. End users can use the `mkdir` command to create new directories, the `cd` command to change directories, and the `ls` command to list directory contents. AdvFS logical structures, quota controls, and backup capabilities are based on traditional file system design. AdvFS replaces or eliminates several standard commands, such as `newfs`, `dump`, `restore`, and `fsck`. AdvFS commands and utilities, and a comparison of AdvFS and UFS commands are detailed in Appendix B.

Without taking an AdvFS system off line, system administrators can backup, reconfigure, and tune file systems. End users can retrieve their own unintentionally deleted files from predefined trashcan directories or from AdvFS fileset clones without assistance from system administrators.

AdvFS supports multivolume file systems, which enables file-level striping (spreading data to more than one volume) to improve file transfer rates for applications that require intensive I/O. **Logical Storage Manager (LSM)**, which allows volume-level striping, can be incorporated into AdvFS configurations.

1.2 License Registration

AdvFS is the file system default on the operating system. AdvFS Utilities is a separately licensed product that provides additional processing capabilities. Before you can use the file system utilities, you must register a license **product authorization key (PAK)** for AdvFS Utilities. Contact your software support organization for additional information.

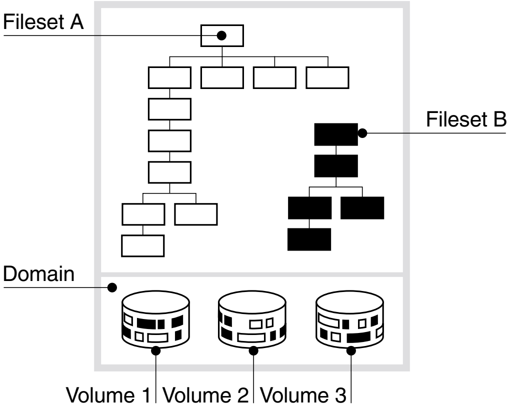
1.3 File System Design

The AdvFS file system consists of two distinct layers: the directory hierarchy layer and the physical storage layer. The directory hierarchy layer implements the file-naming scheme and POSIX-compliant functions such as creating and opening files, or reading and writing to files. The physical storage layer implements write-ahead logging, caching, file allocation, and physical disk I/O functions.

The decoupled file system structure enables you to manage the physical storage layer apart from the directory hierarchy layer. This means that you can move files between a defined group of disk volumes without changing pathnames for your files. Because the pathnames remain the same, the action is completely transparent to end users.

AdvFS implements two unique file system concepts: **filesets** and **domains**. Filesets and domains enable the two-layer file system structure. Figure 1–1 is a representation of this structure.

Figure 1–1: AdvFS File System Design



AdvFS can incorporate Logical Storage Manager (LSM) volumes into the file system structure. When AdvFS is configured with LSM, file system reliability and availability improve because AdvFS can take advantage of LSM features (see Section 1.10).

1.3.1 Filesets, Domains, and Volumes

A fileset follows the logical structure of a traditional UNIX file system. It is a hierarchy of directory names and file names, and it is what you mount on your system. AdvFS goes beyond the traditional file system by allowing you to create multiple filesets that share a common pool of storage called a domain. See Section 1.7 for more information about filesets.

A domain represents the physical storage layer. It is managed separately from the directory structure. You can add or remove volumes without affecting the directory structure. See Section 1.6 for more information about domains.

A **volume** is any mechanism that behaves like a UNIX block device: an entire disk, a disk partition, or an aggregate volume provided by a logical storage manager (see Section 1.5). When first created, all domains consist of a single volume. If you have the optional AdvFS Utilities (see Section 1.2), you can transform a single-volume domain into a multivolume domain by adding one or more volumes.

1.3.2 Transaction log file

The AdvFS file system is a log-based file system that employs **write-ahead logging** to ensure the integrity of the file system. Modifications to the **metadata** (file system structures) are completely written to a **transaction log file** before the actual changes are written to disk. The contents of the transaction log file are written to disk at regular intervals.

When you create a domain, AdvFS creates a transaction log file for it. During crash recovery, AdvFS reads the transaction log file to confirm file system transactions. All completed transactions are committed to disk and uncompleted transactions are undone. The number of uncommitted records in the log, not the amount of data in the file system, dictates the speed of recovery. This means that recovery usually takes only a few seconds. Traditional UNIX file systems rely on the `fsck` utility to recover from a system failure. The `fsck` utility can take hours to check and repair a large file system.

By default, only file system structures are logged, but you can choose to log file data and change the way your system writes to storage (see Section 4.4). A file that has data logging turned on remains internally consistent in the event of a system crash. However, enabling data logging can slow system performance.

1.3.3 File Storage Allocation

Files are not static; their space requirements change over time. To maintain contiguous file placement without overallocating space on the disk, AdvFS uses a unique file storage allocation scheme.

Key features of file storage allocation are:

- Extents

  An **extent** is a contiguous area of disk space that AdvFS allocates to a file. Extents are composed of sets of one or more 8 KB **pages**. When storage is added to a file, it is grouped in extents. There is an **extent map** for each volume on which the file system resides.

  File I/O is most efficient when there are few extents. If a file consists of many small extents, it takes more I/O processing to read or write that file.

  Given the dynamic nature of a file system, the file storage allocation cannot always guarantee contiguous placement of pages. The following factors affect placement:

  - Excessive disk fragmentation

    When a disk is fragmented there are many small free spaces, so AdvFS writes data to isolated physical pages, based on availability, instead of writing to contiguous pages. This might result in files with many extents.

  - Multiple users

    When there are many users on a system, requests for space increase, decreasing the likelihood of contiguous file allocation.

  If you have a domain with files containing many extents, you can decrease the number by running the `defragment` utility (see Section 4.8).

- Preallocation

  Each time a file is appended, AdvFS adds pages to the file by preallocating one-fourth of the file size, up to 16 pages. Excess preallocated space is truncated when the file is closed.

  For multivolume domains, new files are allocated sequentially across volumes. Volumes that are more than 86% full (allocated) are not used for new file allocation unless all volumes are more than 86% full. When data is appended to existing files, storage is allocated on the volume on which the file was initially allocated until the volume is full.

- Fragments

  AdvFS writes files to disk in sets of 8 KB pages. In files where holding the last bytes in an 8 KB page would represent a waste of more than 5% of the allocated space, a **file fragment** is created. The fragment, which ranges in size from 1 KB to 7 KB, is allocated from the fileset's **frag file**. This technique considerably reduces the amount of unused, wasted disk space but might affect file system performance (see Section 4.2).

- Sparse files

  A **sparse file** is one that does not have disk storage reserved for some of its pages. Reading a sparse file at a page that does not have storage reserved returns zeroes. Writing to a page that does not have storage reserved allocates a page of disk storage.

  You can create a sparse file by setting a file's end-of-file using the `ftruncate` command or by using the `lseek` and `write` commands to skip over unwritten sections of the file.

  Core files are sparse files. They have large areas with no information and do not use disk blocks for locations where no data exists. Quota files are sparse because they are indexed by user ID. If there are gaps in the user IDs, there might be sections of the file with no data.

  In contrast, database applications generally reserve storage for the entire file even when data is not available. The applications generally write zeroes in the pages that have no useful data. By writing the data sequentially, the database application creates a database file with large sets of contiguous pages and a small number of extents.

To examine the length of a sparse file, including the pages that do not have disk storage, use the `ls -l` command. The `ls -s` command displays the amount of storage actually used by the file. The `du` command provides the same information as the `ls -s` command (see Section 5.1).

1.4 Setting Up an AdvFS File System

When planning your configuration, consider setting up the `/` (root) and `/usr` file systems on AdvFS. By using AdvFS for root and `/usr`, you increase configuration flexibility and significantly reduce system down time in the event of a system failure.

You can set up AdvFS to resemble a traditional UFS configuration with one partition (volume) per domain and one fileset in each domain. If you have the optional AdvFS Utilities, when you need space, you can add volumes (except to the local root which is restricted to one volume) to increase the size of the domain without changing anything in the existing configuration.

An active AdvFS file system requires one domain and one mounted fileset.

To create an active domain:

1. Create a domain and assign a volume to it (see Section 1.6.5).
2. Create a fileset (see Section 1.7.4).
3. Create a mount-point directory (see Section 1.7.6).
4. Mount a fileset (see Section 1.7.6).

You can create one fileset per domain or you can create many. See Section 1.6.1 and Section 1.7.1 for guidelines.

AdvFS is fully supported in the `/etc/fstab` file to automatically mount a fileset at system boot (see Section 1.7.2). Alternately, you can use a graphical interface or the SysMan Menu to perform this activity.

See Appendix B for a complete list of AdvFS commands. For more detailed information on AdvFS configuration, see *System Configuration and Tuning*.

1.5 Volumes

An AdvFS volume can be a raw disk partition, an entire disk, an aggregate volume provided by LSM, or a hardware or software redundant array of independent disks (RAID) storage.

Each volume in an AdvFS domain contains a **bitfile metadata table (BMT)**, which stores file data structure (metadata), and a **storage bitmap**, which tracks free and allocated disk space. In addition, on one volume each domain has a transaction log file, which stores all metadata changes until the changes are written to disk.

A volume can be assigned to only one domain. It is associated with its domain by a **domain ID**, which is automatically stored in the domain-attributes record on the volume.

If you have the optional AdvFS Utilities, you can add volumes to create a multivolume domain. Multivolume domains increase the storage available for the filesets and allow for preventative disk maintenance. You can add volumes immediately after creating the domain, even before creating and mounting filesets. To perform preventative disk maintenance, you can add a new volume to the domain, migrate your files to the new volume, and then remove the old volume.

For AdvFS to function properly, the number of volumes in a domain with the same domain ID must remain consistent with the number of volumes stored in the domain-attributes record. In addition, each domain is defined by an entry in the `/etc/fdmns` directory (see Section 1.6.2). This directory must be up to date; that is, the domain entries must correctly reference the volumes associated with the domains. The number of links to the volumes in the `/etc/fdmns` directory must equal the number of volumes.

To reuse a volume see Section 5.8.1.

1.5.1 Volume Attributes

You can configure AdvFS volumes with attributes that determine how data is read, cached, written, and consolidated. When an AdvFS volume is incorporated into a domain, either by creating the initial domain or by adding a volume, the default volume attributes are set. Modifying these default attributes might improve performance in some system configurations.

You can run the `chvol` command Manage AdvFS Domains at any time to change the attributes of a volume in an active domain; the system does not have to be quiescent.

To display or modify the current volume attributes, use the SysMan Manage an AdvFS Domain utility, or enter the `chvol` command from the command line:

```
chvol device_name file_domain
```

See Chapter 4 and *System Configuration and Tuning* for more complete information on modifying attributes to improve system performance.

1.6 Domains

A domain is the physical storage layer of the AdvFS file system. It is a defined pool of physical storage that can contain one or more volumes. Because this storage is managed separately from the directory structure (see Section 1.6.2), you can expand and contract the size of the domain by adding or removing volumes. You can move files between volumes in the domain without changing file pathnames. Changing the name of the domain does not affect the domain ID.

The domain name is defined by a directory entry in the /etc/fdmns directory. This entry contains the links to the volumes that comprise the domain.

Domain names must be unique. You cannot use slash (/), number (#), colon (:), asterisk (*), question mark (?), tab, newline, form feed, return, vertical tab, and space characters in a domain name.

1.6.1 Configuring Domains

How you configure domains depends on your organization's needs. You can assign all available storage to a few domains or you can group specific partitions or disks into many domains. You must also take into account how you configure your filesets (see Section 1.7.1). With AdvFS Utilities, you can assign multiple volumes to a domain and distribute the filesets and files across the volumes.

Establishing multiple domains allows greater control over your physical resources. You can create domains for use by specific projects, groups of users, departments, or any division that makes sense for your organization. For example, you could create domains for each of your organization's departments, such as engineering, finance, and personnel.

There are a number of factors to consider when configuring domains:

- To maintain high performance, do not split a disk between two domains. For example, do not add partition g to one domain and partition h of the same disk to another domain.
- Adding one partition containing the entire disk (typically, partition c) to a domain is preferable to adding several partitions on the same disk. Adding several partitions might degrade performance. An error message is displayed if you try to assign overlapping partitions to a domain (see Section 1.6.6).
- Assuming the disks are the same type and speed, it is generally more efficient to spread your domain over several disks. For example, a domain with three volumes on separate disks is more efficient than one with three partitions on a single disk because the latter has only one I/O path.

- Creating a domain on several volumes can increase the defragmentation speed because the `defragment` utility processes each volume in parallel.
- Combining multiple volumes within a single domain allows you to build larger domains and reduces the overall management effort because fewer domains require less administration. However, a single volume failure within a domain renders the entire domain inaccessible.
- Limiting the number of volumes decreases the risk of disk errors that can cause the entire domain to become inaccessible. To improve reliability, you can set up LSM **mirror**s. See Section 1.10 for information about setting up AdvFS with LSM. See *Logical Storage Manager* for LSM functionality.

The *System Configuration and Tuning* guide provides detailed guidelines for planning and configuring your file system. *System Limits* lists the limits for the AdvFS file system. The SysMan Manage an AdvFS Domain utility allows you to create and configure domains. See Appendix D to configure your domains using a graphical user interface.

1.6.2 The /etc/fdmns Directory

The `/etc/fdmns directory` defines domains by providing a subdirectory for each domain you create. The subdirectories contain a symbolic link to every volume in the domain.

AdvFS automatically creates and maintains this directory when you use standard AdvFS commands. You must have a current `/etc/fdmns` directory in order to mount the filesets in a domain.

When you create a domain, a soft link is created from the domain entry in the `/etc/fdmns` directory to the block device. You cannot create a domain by creating a link in the directory.

Back up the `/etc/fdmns` directory regularly. If the contents of the directory become corrupt or if the directory is deleted, restore the directory from your most recent backup. A damaged directory prevents access to the domain because the information matching the domain to the physical volume containing the filesets is incorrect; the filesets are not affected.

If you attempt to mount a fileset from a domain with a damaged directory, a message similar to the following for the domain `accounts` is displayed:

```
Volume count mismatch for domain accounts.
accounts expects 2 volumes, /etc/fdmns/accounts has 1 links.
```

If the `/etc/fdmns` directory is damaged or if the volumes of a domain are moved to a different system, use the `advscan` utility to locate the volumes and reconstruct the directory (see Section 5.9).

1.6.3 Version 5.0 Domains

Domains in Version 5.0 and later of the operating system have an improved disk structure that provides support for quota values larger than 2 terabytes and increases performance for directories containing thousands of files. You do not need to adjust bitfile metadata table (BMT) allocations. All domains that were created in operating systems prior to Version 5.0 are recognized by later versions but are not automatically upgraded to the new structure.

A **domain version number (DVN)** is associated with a domain. The Version 5.0 operating system is the first for which a new domain carries a DVN of 4. Domains created earlier carry a DVN of 3.

You can mount a DVN3 fileset on a system running Version 5.0 or later. You cannot locally mount a fileset in a DVN4 domain created under Version 5.0 or later on a system running a Version 4 operating system. If you need to access a DVN4 fileset while running an earlier version of the operating system, NFS mount it from a server running Version 5 or later software (see Section 5.4). Therefore, if your application requires backward compatibility, you should not upgrade your domain to the new DVN.

See Section 5.4.1 for a discussion of utility incompatibility.

1.6.3.1 Creating a Domain with DVN4

All domains created on Version 5.0 and higher operating systems have a DVN of 4. Therefore, if you do a full installation, all the domains created in the process have the new structure.

If you perform an update installation from a system running a Version 4 operating system, all existing domains retain the DVN of 3. This means that `/root`, `/usr`, and `/var` also have a DVN of 3. There is no conversion utility. To move your data to a DVN4 domain, back up your data, create a new DVN4 domain, and restore it to the new domain.

1.6.3.2 Upgrading a Domain to DVN4

If you are running an application that requires quota limits larger than 2 terabytes or that uses directories containing thousands of files, you can improve performance by upgrading your old domain to a DVN4 domain. DVN4 domains automatically have an index created when the directory grows beyond a page, that is, about 200 files. Use the `showfile` command to determine if a directory is indexed. Use the `-i` option to display information about the index. See `showfile`(8) for more information.

To upgrade a domain, create a new domain on a system running Version 5.0 or later software and copy all the information from the old domain to

it. If you are unfamiliar with creating domains, read Section 1.6.5 first.
To upgrade a domain:

1. Back up the filesets in the domain to tape by using the `vdump` command. It is a good idea to use the `-x` option for additional protection from saveset errors. You lose all the data in your domain if you cannot restore it after creating a new domain.
2. Remove the old domain by using the `rmfdmn` command.
3. Create a new domain by using the `mkfdmn` command. The new domain has a DVN of 4. You do not need to use the `-x` and `-p` options for the `mkfdmn` command. DVN4 domains do not require additional of BMT allocations.
4. Create the new filesets by using the `mkfset` command.
5. Restore the filesets in the new domain by using the `vrestore` command.

For example, to upgrade the domain `domain_p` on `/dev/disk/dsk1c` containing filesets `fset_p` and `fset_m` and put them back on the same volume with the same names, enter the following commands:

```
# vdump -0 -N -x 8 /fset_p
# vdump -0 -N -x 8 /fset_m
# umount /fset_p
# umount /fset_m
# rmfdmn domain_p
# mkfdmn /dev/disk/dsk1c domain_p
# mkfset domain_p fset_p
# mkfset domain_p fset_m
# mount domain_p#fset_p /fset_p
# mount domain_p#fset_m /fset_m
# vrestore -x -D /fset_p
# mt fsf 1
# vrestore -x -D /fset_m
```

You can create DVN3 domains on a Version 5 system. They are recognized by earlier versions of the operating system. See `mkfdmn`(8) for more information.

1.6.4 Displaying Domain Information

If a domain is active (at least one fileset is mounted), you can display detailed information about the domain and the volumes included in it by using the `showfdmn` command:

showfdmn *domain_name*

For example, to display domain information for the `domain_1` domain:

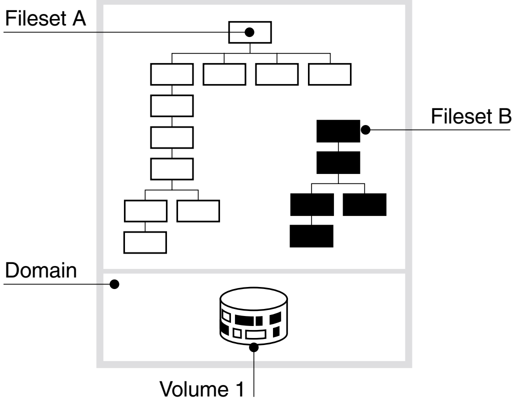
```
# showfdmn domain_1
      Id           Date Created      LogPgs Version Domain Name
2bb0c594.00008570 Fri Mar 24 12:33 2000 512       4 domain_1
Vol  512-Blks  Free % Used Cmode Rblks Wblks Vol Name

  1L     8325 79210    90%    on   128   128 /dev/disk/dsk1c
  2    832527  1684    98%    on   128   128 /dev/disk/dsk2c
      -------------------------
       1665054 80894   94%
```

1.6.5 Creating a Domain

The first step in setting up an AdvFS file system is creating a domain and assigning an initial volume to it. However, a domain is not a complete file system that you can mount. In order to mount an AdvFS file system, the domain must contain one or more filesets. You can access files as soon as you mount one or more filesets (see Section 1.7).

Figure 1–2: Single-Volume Domain



Creating a single-volume domain with a single fileset is equivalent to creating a traditional UFS file system. To set up an active, single-volume file system, as illustrated in Figure 1–2, use the SysMan Create a New AdvFS Domain utility, a graphical user interface (see Appendix D), or enter the `mkfdmn` command from the command line:

```
mkfdmn volume_name domain_name
```

To create an active single-volume DVN4 domain:

1. Create a single-volume domain associated with a volume by using the `mkfdmn` command.

2. Create one or more filesets by using the mkfset command.
3. Create the mount-point directory by using the mkdir command. It is convention to name the directory the same as the fileset. For example, if the fileset is tmp, name the mount-point directory /tmp.
4. Mount each fileset by using the mount command.

For information about creating a DVN3 domain, see mkfdmn(8).

Caution

Do not use the mkfdmn command on a volume containing data that you want to keep. Doing so destroys the data. If you have accidentally used the mkfdmn command, you might recover some of your data by using the salvage utility if the volume is an AdvFS volume (see Section 5.8.4).

The following example creates a single-volume DVN4 domain, domain_2, and two filesets, fset_a and fset_b, in the domain. Because the domain has only one volume, the files in both filesets physically reside on one volume. This is allowed in AdvFS. The number sign (#) between the domain and fileset names is part of the naming syntax and does not represent a comment.

```
# mkfdmn /dev/disk/dsk2c domain_2
# mkfset domain_2 fset_a
# mkfset domain_2 fset_b
# mkdir /fset_a
# mkdir /fset_b
# mount domain_2#fset_a /fset_a
# mount domain_2#fset_b /fset_b
```

You can set up a domain with an LSM storage by specifying an LSM volume. See Section 1.10 for information about using AdvFS with LSM. If you have AdvFS Utilities, you can change the size of your domain by adding more volumes. You can transform a single-volume domain (except the root domain) into a multivolume domain (see the Sysman Manage an AdvFS Domain utility, Section 1.6.6 and Appendix D).

If you try to create a domain on a volume that is marked on the disk as in use in the disk label, the mkfdmn command fails if the volume is mounted or if it is a current swap partition. If the volume is unmounted, you get an warning message. You can override the message and create the domain. For example, if /dev/disk/dsk3g is in use and you try to create the domain usr_domain, the following message is displayed:

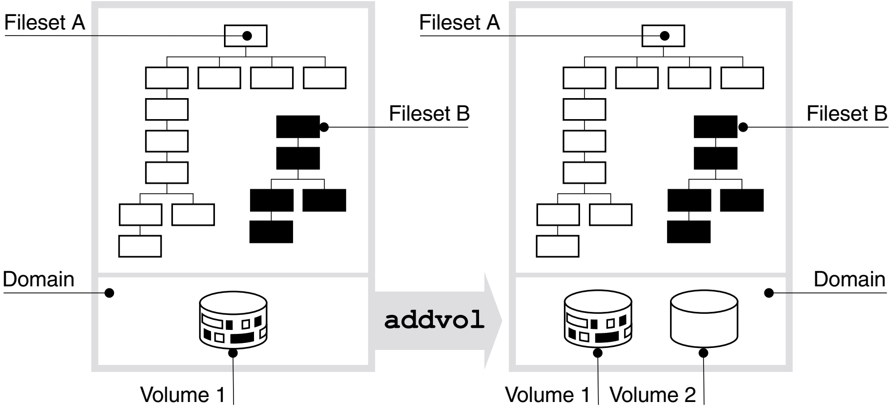
```
# mkfdmn /dev/disk/dsk3g usr_domain
Warning: /dev/rdisk/dsk3g is marked in use for 4.2BSD.
```

```
If you continue with the operation you can
possibly destroy existing data.
CONTINUE? [y/n] <y>
```

1.6.6 Increasing the Size of a Domain

You can expand a domain by replacing one of the volumes in the domain with a larger storage device or, if you have AdvFS Utilities, by adding another volume to the domain. Figure 1–3 shows a graphic illustration of adding volumes to a domain. Neither adding nor removing volumes affects the directory hierarchy layer; all pathnames for the files remain the same. Also, the file system can remain active during the disk exchange.

Figure 1–3: Enlarging a Domain



There might be performance benefits if, instead of increasing the size of a DVN3 domain, you upgrade to the new file structure (see Section 1.6.3) and then increase its size.

Caution

If your domain is located on an LSM volume, do not use the LSM `grow` option to increase its size.

A newly created domain consists of one volume, which can be a disk, disk partition, or logical volume. To add volumes, use the SysMan Manage an AdvFS Domain utility, a graphical user interface (see Appendix D), or enter the `addvol` command from the command line:

```
addvol device_name domain_name
```

For example, to add volume `dsk3c` to the domain `resources`:

```
# addvol /dev/disk/dsk3c resources
```

You can add volumes immediately after creating a domain, or you can wait until the domain requires additional space. You can add a volume to an active domain while its filesets are mounted and in use.

You cannot add a volume to the root domain unless you are running a cluster (see *Cluster Administration*). If you are running a cluster configuration, you add another root volume the same way that you add any volume. If you are not running a cluster and need to increase the size of the root domain, follow the procedure for recovering from a failure of the root domain (see Section 5.11) and attach a larger volume.

Caution

> Do not use the `addvol` command to add a volume containing data that you want to keep. Doing so destroys the data. If you have accidentally used the `addvol` command, you might recover some of your data by using the `salvage` utility (see Section 5.8.4).

If you do not have AdvFS Utilities and want to increase the size of your domain:

1. Make a new domain on the new larger device. Do not use the same name for the domain.
2. Create filesets with the same name as the old filesets.
3. Create a temporary mount-point directory for each fileset.
4. Mount each new fileset on its temporary mount point.
5. Use a utility (for example `vdump/vrestore, cpio, cp -R, or tar`) to copy each fileset from the old device to the newly mounted fileset.
6. Unmount the old and new filesets.
7. Rename the new domain to the old name if you want. If you do not change the domain and fileset names, you do not need to to edit the `/etc/fstab` file. If you do change the names, change them in the `/etc/fstab` file.
8. Mount each new fileset using the mount point of the old fileset. The directory tree is then unchanged. Delete all temporary mount-point directories.

To increase the size of a domain if you have AdvFS Utilities:

1. Optionally, use the `showfdmn` command to display the contents of the domain and the current disk capacity of each volume.
2. Add the new volume to the domain.
3. Remove the old volume if you do not want it.

4. Optionally, run the balance utility to even the file distribution between the volumes.

The following example replaces one disk, the volume /dev/disk/dsk2c, of the domain_1 domain with a larger disk, the volume /dev/disk/dsk3c:

```
# showfdmn domain_1
    Id              Date Created     LogPgs Version Domain Name
2bb0c594.00008570 Fri Mar 24 12:33 2000 512       4 domain_1

Vol  512-Blks  Free % Used Cmode Rblks Wblks Vol Name
  1L   832527 79210    90%    on   128   128 /dev/disk/dsk1c
  2    832527  1684    98%    on   128   128 /dev/disk/dsk2c
     -------------------------
      1665054 80894    94%
# addvol /dev/disk/dsk3c domain_1
# rmvol /dev/disk/dsk2c domain_1
# balance domain_1
```

AdvFS does not add a volume that causes partitions to overlap with a volume that is mounted for another file system, a swap area, or a reserved partition. To add an overlapping partition, unmount all filesets and use the addvol -F command.

Caution

Disabling the overlap check can result in extensive data loss and should be used with extreme caution (see addvol(8)).

1.6.7 Reducing the Size of a Domain

When there is sufficient free space on the remaining volumes, you can remove volumes from a domain without interrupting users or affecting the logical structure of the filesets in the domain. The file system automatically migrates the contents of the selected volume to other volumes in the domain. Before you can remove a volume from a domain, all filesets in the domain must be mounted. You get an error if you try to remove a volume from a domain with unmounted filesets.

If there is not enough free space on other volumes in the domain to accept the files that are offloaded from the departing volume, as many files as possible are moved to available free space on other volumes. Then, a message is sent indicating that there is insufficient space. The domain is not damaged.

Caution

If your domain is located on an LSM volume, do not use the LSM `shrink` option to reduce its size.

To remove a volume, use the SysMan Manage an AdvFS Domain utility, a graphical user interface (see Appendix D), or enter the `rmvol` command from the command line:

```
rmvol device_name domain_name
```

For example, to remove `dsk3c` from the domain `inventory`:

```
# rmvol /dev/disk/dsk3c inventory
```

You can interrupt the `rmvol` process (see `rmvol(8)`) with Ctrl/c or by using the `kill -term` command without damaging your domain. Files already removed remain in their new location. Do not use the `kill -KILL` command.

Under some circumstances interrupting an `rmvol` command by using the `kill` command can leave the volume in an inaccessible state; that is, it cannot be written to. Volumes in this condition are marked as "data unavailable" in the output of the `showfdmn` command. If a volume does not allow writes after an aborted `rmvol` operation, use the `chvol -A` command to reactivate the volume.

To reduce the size of a domain:

1. Use the `showfdmn` command to display the contents of the domain and the current disk capacity of each volume. This step is optional.
2. Remove the volume.
3. Run the `balance` utility to even the file distribution between the remaining volumes. This step is not required.

You cannot remove a volume from a domain that you are balancing or defragmenting.

The following example illustrates how to remove one disk of the `domain_2` domain, `/dev/disk/dsk2c`:

```
# showfdmn domain_2
  Id              Date Created     LogPgs Version Domain Name
2bb0c594.00008570 Fri Jun 9 10:23 2000 512      4 domain_2
 Vol 512-Blks   Free % Used Cmode Rblks Wblks  Vol Name

 1L   832527  386984    54%    on   128   128 /dev/disk/dsk1c
 2    832527  647681    22%    on   128   128 /dev/disk/dsk2c
 3    832527  568894    32%    on   128   128 /dev/disk/dsk3c
```

```
      ----------------------
      249758 1603559    36%
# rmvol /dev/disk/dsk2c domain_2
# balance domain_2
```

If you remove an AdvFS volume that contains a stripe segment, the rmvol utility moves the segment to another volume that does not already contain a stripe segment of the same file. When a file is striped across all volumes in the domain, a confirmation is required before removing the volume. If you allow the removal process to continue, more than one stripe segment is placed on the remaining volumes. See Section 4.13 for details on file striping.

1.6.8 Removing a Domain

You can remove a domain after all filesets in the domain are unmounted. When you remove a domain, the entry in the /etc/fdmns directory that defined the domain is removed and you cannot mount the filesets. Volumes that were assigned to the removed domains are relabeled as unused and can be reused.

Removing the domain to inactivate filesets instead of using the rmfset command to remove each fileset might present a security hole because no data on the volumes is changed. You might be able to access your data by using the salvage command (see Section 5.8.4).

To remove a domain, use a graphical user interface (see Appendix D), or unmount all filesets and fileset clones from the command line. Then, enter the rmfdmn command:

```
rmfdmn domain_name
```

For example, to remove the domain promotions:

```
# rmfdmn promotions
rmfdmn: remove domain promotions? [y/n]y
rmfdmn: domain promotions removed
```

If you use this command when there are mounted filesets, the system displays an error message. AdvFS does not remove an active domain.

1.6.9 Renaming a Domain

You can assign a new name to an existing domain without altering its domain ID. When you rename a domain, entries for all filesets in the domain must be updated in the /etc/fstab file (see Section 1.7.2). To rename a domain, use a graphical user interface (see Appendix D) or, from the command line:

1. Unmount all the filesets and any related clones.

2. In the /etc/fdmns directory, change the old domain name to the new one:

   ```
   #mv /etc/fdmns/old_dom_name /etc/fdmns/new_dom_name
   ```

3. Edit the /etc/fstab file to enter the new domain name and remove the old.
4. Mount the filesets in the renamed domain.

For example, to rename the domain marketing to advertising, assuming one fileset, fset, is mounted at /fset using the vi editor:

```
# umount /fset
# mv /etc/fdmns/marketing /etc/fdmns/advertising
# vi /etc/fstab
```

Locate the following in the /etc/fstab file:

```
marketing#fset /fset advfs rw,userquota,groupquota 0 2
```

Edit this line as follows:

```
advertising#fset /fset advfs rw,userquota,groupquota 0 2
```

Mount the fileset:

```
# mount /fset
```

1.7 Filesets

A fileset represents a mountable portion of the directory hierarchy of a file system. Filesets and traditional UNIX file systems are equivalent in many ways. You mount AdvFS filesets. Filesets contain files, are units on which you enable quotas, and are units for backing up data.

In contrast with traditional file systems, the directory hierarchy of AdvFS is independent of the storage. Therefore, you can change file placement without affecting the logical structure of the filesets.

Fileset names can be associated with their domain names, as in domain#fileset. Here the number sign (#) is part of the naming syntax and does not represent a comment.

Fileset names must be unique within a domain. Each fileset has a unique **fileset ID** composed of the domain ID and a fileset tag. Each domain has its own series of fileset tags.

AdvFS also supports **fileset clones**. An AdvFS fileset clone is a read-only copy of an existing fileset created to capture data at one instant in time.

Fileset names must be unique. You cannot use slash (/), number (#), colon (:), asterisk (*), question mark (?), tab, newline, form feed, return, vertical tab, and space characters. Filesets that are not in the same domain can have the same name.

1.7.1 Configuring Filesets

The filesets in a domain share the available space on the volumes in the domain and use the same domain transaction log file. Each fileset has its own directory structure, root tag directory, quota files, and frag file. The optimal number of AdvFS filesets in a domain depends primarily on the requirements of the applications that use the filesets.

It is generally a good idea to have multiple filesets rather than one large fileset. Whether you place these filesets in a number of domains rather than in one large domain depends on your application. If a fileset has critical access requirements, place it in its own domain. For applications that perform many file create and remove operations, configure multiple domains, each on its own volume. This reduces contention related to I/O.

Create multiple smaller filesets rather than a few large filesets if you do not want to share space between filesets. You can then set fileset quotas to limit resource use for each fileset.

Create filesets containing files having similar requirements in the same domain. For example, do not place small temporary mail news server files in the same domain as the database.

Because filesets are managed independently of their physical storage, each fileset can be backed up independently (see Chapter 2) and can be assigned its own quota limits (see Chapter 2). Multiple small filesets can be backed up and restored more quickly than a single large fileset. You can run the `vdump` or `vrestore` commands on several filesets simultaneously.

You can group files in filesets by their management requirements. For example, you can create a fileset for developers' files that is backed up twice a day and you can create another fileset with quotas imposed to limit the amount of disk space available to the marketing department.

All filesets share the transaction log file in a domain, so filesets with a large amount of I/O can cause the transaction log file to become a bottleneck (see Section 1.3.2). Balance the management gains of having multiple filesets in a domain against the potential performance reduction you might incur by having all of the log data for all filesets going to one transaction log file (see Section 4.3).

You can create and mount filesets until the system runs out of system resources (such as memory or disk space). *System Limits* contains information about this and other system limits.

System Configuration and Tuning provides detailed guidelines and suggestions for file system configuration. The SysMan Manage an AdvFS Domain utility allows you to create and configure filesets. You can use a graphical user interface to accomplish many configuration operations (see Appendix D).

1.7.2 Designating Filesets in the /etc/fstab File

Add AdvFS filesets to the `/etc/fstab` file by listing them with an `advfs` designation (see `fstab`(4)). This is similar to the way that you add any other file system. AdvFS filesets listed in the `/etc/fstab` file are mounted each time you reboot the system.

The fileset entry includes the domain name, fileset name, mount point, file system type, and the mount-point options. If you want to enable quotas, include the `userquota` and `groupquota` options (see Section 2.2) and the pass field numbers:

```
file_dmn#fileset /mount_point advfs rw,userquota,groupquota 0 2
```

For example, to automatically mount the `credit` fileset (assuming the fileset's mount point exists), add the following line to the `/etc/fstab` file:

```
acct_domain#credit /credit advfs rw,userquota,groupquota 0 2
```

The `userquota` and `groupquota` options identify the mounted file systems that can be processed by quota-related commands. Many quota commands have a `-a` (all) option that directs the command to operate only on file systems specified in this way.

1.7.3 Displaying Fileset Information

Any system user can display detailed information about mounted filesets and clones. Root user privilege is required only if the domain is inactive (filesets unmounted). To examine fileset information enter:

`showfsets` *`domain_name`*

The following example displays the domain `big_domain`, which has four filesets:

```
# showfsets big_domain
staff1_fs
    Id           :  2cb9d009.000419f4.1.8001
    Files        :   18554,  SLim= 0, HLim= 0
    Blocks(512)  :  712230,  SLim= 0, HLim= 0
    Quota Status :  user=on group=on
    Object Safety: off
    Fragging     : on
    DMAPI        : off
```

```
 guest_fs
    Id           :  2cb9d009.000419f4.2.8001
    Files        :    4765,  SLim= 0, HLim= 0
    Blocks(512)  :  388698,  SLim= 0, HLim= 0
    Quota Status :  user=on group=on
    Object Safety: off
    Fragging     : on
    DMAPI        : off
 staff2_fs
    Id           :  2cb9d009.000419f4.3.8001
    Files        :   12987,  SLim= 0, HLim= 0
    Blocks(512)  :  842862,  SLim= 0, HLim= 0
    Quota Status :  user=on group=on
    Object Safety: off
    Fragging:      on
    DMAPI        : off
 staff3_fs
    Id           :  2cb9d009.000419f4.4.8001
    Files        :   48202,  SLim= 0, HLim= 0
    Blocks(512)  : 1341436,  SLim= 0, HLim= 0
    Quota Status :  user=on group=on
    Object Safety: off
    Fragging     : on
    DMAPI        : off
```

The following example displays domain_2, which contains one fileset and one fileset clone:

```
# showfsets domain_2
test_fs
    Id           :  3003f44f.0008ac95.4.8001
    Clone is     : clone_test
    Files        :    7456,  SLim= 0, HLim= 0
    Blocks (512) :  388698,  SLim= 0, HLim= 0
    Quota Status : user=on group=on
    Object Safety: off
    Fragging     : on
    DMAPI        : off
 Clone_test
    Id           : 3003f44f.0008ac95.5.8001
    Clone of    : test_fs
    Revision    : 2
```

You can use a graphical user interface (see Appendix D) to obtain similar information.

1.7.4 Creating a Fileset

A domain must contain at least one mounted fileset to be active (see Section 1.3.1). Within a domain you can create multiple filesets that share the storage pool established for the domain. Any fileset can consume all of the storage available in the domain.

Each fileset can be mounted and unmounted independently of the other filesets in the domain. You can limit fileset growth within a domain by assigning fileset quotas (see Chapter 2). You can control the creation of frags for files that waste more than 5% of their allocated storage (see Section 4.2). You can set the characteristics of the fileset at creation by using the `mkfset` command or change them by using the `chfsets` command (see Section 1.7.10).

To create a fileset in a domain, use the SysMan Create a New AdvFS Fileset utility, a graphical user interface (see Appendix D), or enter the `mkfset` command from the command line:

```
mkfset domain_name fileset_name
```

To mount a fileset, you must create a mount-point directory for it (see Section 1.7.6).

For example, to create the fileset `coupons` in the domain `advertising`:

```
# mkfset advertising coupons
```

See also Section 1.6.5, which contains examples for configuring an active (filesets mounted) domain.

1.7.5 Upgrading a Fileset

Filesets that are part of domains created for operating system software Version 5.0 and later (DVN4) support large quota values and have better performance for very large directories. Filesets in domains created earlier (DVN3) do not have these improvements. To upgrade a fileset to the new version, you must upgrade its domain (that is, create a new one) and restore the fileset to it. Then the restored fileset has the new quota limits and performance improvements (see Section 1.6.3).

You cannot mount filesets with the new DVN on operating system versions earlier than Version 5.0 unless you NFS mount them from a server running Version 5.0 or later of the operating system software (see Section 5.4).

1.7.6 Mounting a Fileset

As with traditional UNIX file systems, AdvFS filesets must be mounted in order to access them. You must create a mount-point directory if it does not exist. Filesets to be mounted must be compatible with the operating system on which they were created (see Section 5.4).

To mount a fileset, use the SysMan General File System Utilities - Mount File Systems utility, a graphical user interface (see Appendix D), or enter the mount command from the command line:

```
mount domain_name#fileset_name mnt_point
```

For example, to create a mount-point directory and mount the fileset coupons in the domain advertising:

```
# mkdir /coupons
# mount advertising#coupons /coupons
```

Before a fileset is mounted, AdvFS verifies that all data in all volumes in a domain can be accessed. If there are problems, the mount might fail or the fileset might be mounted as read-only (see Section 5.8.7).

If you attempt to mount a fileset with an incorrect number of volumes, the mount operation fails. See Section 5.9.2 and advscan(8) for further information.

You can specify temporary atomic write data logging in your mount operation (see Section 4.4).

An error message is displayed if you attempt to mount a fileset created under operating system software Version 5.0 or later on a system running Version 4 or earlier operating system software (see Section 5.4).

1.7.7 Unmounting a Fileset

If you unmount a fileset, the fileset remains in the domain but it is not accessible. Mount the fileset to make it available again.

To unmount an AdvFS fileset, use the SysMan General File System Utilities - Dismount a File System utility, a graphical user interface (see Appendix D), or enter the umount command from the command line:

```
umount mnt_point
```

For example, to unmount the fileset coupons in the domain advertising that was mounted in Section 1.7.6:

```
# umount /coupons
```

1.7.8 Removing a Fileset

If you remove a fileset, it cannot be remounted. It is no longer part of the domain.

You must unmount a fileset and remove the fileset clone before you can remove it. If you have set up a trashcan directory (see Section 1.8) for the fileset, it is also removed.

To remove a fileset, use the SysMan Manage an AdvFS Domain utility, a graphical user interface (see Appendix D) or, from the command line, enter the rmfset command:

```
rmfset domain_name fileset_name
```

For example, to remove the tmp_1 fileset in domain_1:

```
# rmfset domain_1 tmp_1
rmfset: remove fileset tmp_1? [Y/N]y
```

The fastest way to remove all filesets is to remove the domain by using the rmfdmn command. However, you might be able to access some of the data by using the salvage utility (see Section 5.8.4). The rmfdmn command removes the definition of the domain in the /etc/fdmns directory and relabels the volumes, but it does not touch any data on the volume. If you remove filesets by using the rmfset command, you cannot recover them by using the salvage utility because the utility destroys pointers to the metadata for all the files in the fileset.

1.7.9 Renaming a Fileset

An unmounted fileset can be renamed. The fileset name is kept within the domain and is an attribute that you assign. When you rename a fileset, only this assignment is changed. The fileset ID is not altered.

To rename a fileset, unmount it and unmount its clone if it has one. You can use the SysMan Manage an AdvFS Domain utility, a graphical user interface (see Appendix D) or, from the command line, enter:

```
renamefset domain_name old_fileset_name new_fileset_name
```

After renaming the fileset, you must update the corresponding entries in the /etc/fstab file (see Section 1.7.2). If you do not do this, AdvFS does not mount the fileset when the system is booted.

For example, to rename the sad fileset that is mounted at /mntpt to happy using the vi editor:

```
# umount /mntpt
# renamefset dmn_1 sad happy
# vi /etc/fstab
```

Locate the following line in the /etc/fstab file:

```
dmn_1#sad /mntpt advfs rw,userquota,groupquota 0 2
```

Edit this line as follows:

```
dmn_1#happy /mntpt advfs rw,userquota,groupquota 0 2
```

Mount the fileset:

```
# mount /mntpt
```

Note

You cannot rename an AdvFS fileset clone. You must delete the old clone and create a new one.

1.7.10 Changing the Characteristics of a Fileset

With the chfsets command you can change the quota limits for a fileset by using the -F, -f, -B, and -b options (see Section 2.2.2) and by using the -o option you can turn fragment files on and off (see Section 4.2), and enable or disable object safety. From chfsets command has the following syntax:

chfsets *[-F limit] [-f limit] [-B limit] [-b limit]*
[-o <frag|nofrag> <objectsafety|noobjectsafety>
<dmapi|nodmapi>] domain_name [fileset_name]

For example, to enable object safety in the xyz3 fileset in the xyz_domain:

```
# chfsets -o objectsafety xyz_domain xyz3
```

When the object safety option is enabled, the pages on disk belonging to the fileset are zero-filled and forced to disk before they are available to the file. This prevents old data from being visible if a system crash occurs while the file is being written.

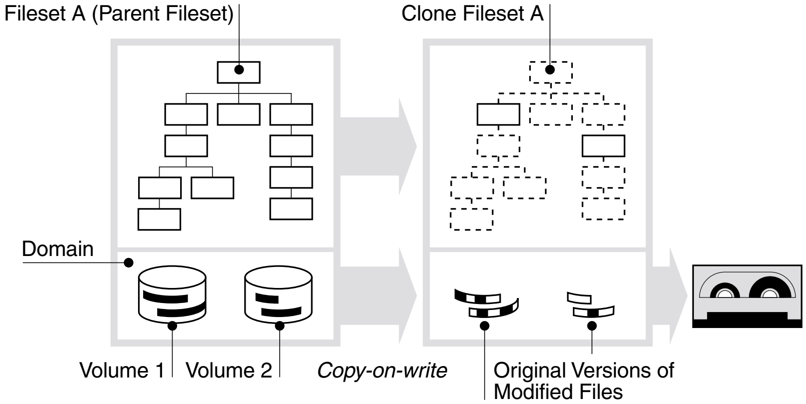
Files that allocated storage just prior to enabling object safety initiate object safety once the buffers associated with the allocation are flushed to disk.

Because object safety degrades performance (the pages on disk are zero-filled then the zeroes are overwritten with data) the administrator must decide if the performance impact is worth the improved security.

1.7.11 AdvFS Fileset Clones

If you have the optional AdvFS Utilities license, you can back up your files using an AdvFS fileset clone. A fileset clone is a read-only snapshot of fileset data structures (metadata). When you clone a fileset (create a fileset clone), the utility copies only the structure of the original fileset, not the actual data. When a file is modified, the file system copies the original, unchanged data to the AdvFS fileset clone. (This is called **copy-on-write**.) Because the only data in the fileset clone is a copy of data that has been modified, the fileset clone is usually smaller than the original fileset. Figure 1–4 illustrates the relationship between parent and fileset clones.

Figure 1–4: Cloning a Fileset



Fileset clones increase the availability of data because they:

- Preserve the system at a particular time

  A clone is not a replacement for the backup process (using the `vdump` command, for example). However, it can provide internal consistency if you are trying to back up a system where files are changing rapidly and you want to retain the information at a particular time. Use the clone to make a snapshot of your data, then use the `vdump` command to back up the clone.

- Protect against accidental file deletion or corruption

  Create a clone of each fileset that you plan to access or modify. By leaving the fileset clone on line, you can replace unintentionally deleted or corrupt files without loading backup tapes.

A clone cannot be cloned.

See Section 3.3 for command-line instructions on using cloning for online backup. See Section D.4.3 for an explanation of cloning with the AdvFS GUI.

Note

> Changing text files with an editor can cause the entire original file to be copied to the clone. Many editors rewrite the entire file regardless of what has changed. When this happens, your AdvFS fileset clone might grow very large. There is no way for AdvFS to alter this process.

When you delete a file that existed when the clone was created, it remains available (but not visible in the original fileset) for the life of the clone. The file is not copied to the clone, but the actual delete operation is delayed until the clone is deleted. The version of the file that is retained is the one that existed when the clone was created. Later updates are lost.

The size of an AdvFS fileset clone depends upon the number of updates that occur during the life of the clone. The `df` command, which displays statistics on free disk space, does not accurately reflect the size of the fileset clone because it constantly changes as files are updated.

Caution

When a domain runs out of disk space, the file system loses its ability to maintain the consistency of files within AdvFS fileset clones. The original fileset is usable, but the fileset clone is not accurate and is no longer used. A warning message is displayed on both the user's terminal and the system console.

To clone a system disk and boot it on another system, see Best Practices. This not the same as creating an AdvFS fileset clone.

1.7.11.1 Creating an AdvFS Fileset Clone

AdvFS fileset cloning is transparent to the user and has little impact on system performance. You must be root user to create a clone. To create a fileset clone, use the SysMan Manage an AdvFS Domain utility, a graphical user interface (see Appendix D), or enter the `clonefset` command from the command line:

```
clonefset domain_name fileset_name clone_name
```

For example, to create a clone called `clone_day300` for the fileset `day300` in the domain `transactions`:

```
# clonefset transactions day300 clone_day300
```

1.7.11.2 Mounting and Unmounting an AdvFS Fileset Clone

Clones are mounted and unmounted in the same manner as any other fileset (see Section 1.7.6 and Section 1.7.7).

1.7.11.3 Removing an AdvFS Fileset Clone

Clones are removed in the same manner as any other fileset (see Section 1.7.8).

1.7.11.4 Renaming an AdvFS Fileset Clone

A fileset clone cannot be renamed. To assign a new name, remove the old clone and create a new clone for the fileset. (Note that this new clone is a snapshot of the fileset at a later point in time than the deleted clone.)

1.8 Setting Up Trashcans

If you have the optional AdvFS Utilities, end users can configure their systems to retain a copy of files they have deleted. They can attach **trashcan** directories to one or more directories within the same fileset. Once attached, any file deleted from an attached directory is automatically moved to the trashcan directory. The last version of a file that was deleted from a directory with a trashcan attached can be returned to the original directory by using the `mv` command.

Trashcan directories are a trade off, however. The convenience of recovering files without accessing backup comes at the cost of the additional writes to disk that are required when files are deleted.

Root user privilege is not required to use this command. However, the following restrictions apply:

- You can restore only the most recently deleted version of a file.
- You can attach more than one directory to the same trashcan directory; however, if you delete files with identical file names from the attached directories, only the most recently deleted file remains in the trashcan directory.
- Only files you delete directly are removed to the trashcan. If you delete a complete fileset using the `rmfset` command, the files in it are not saved.
- Deleted files in an attached trashcan count against your quota.
- When you delete files in the trashcan directory, they are unrecoverable.

Table 1–1 lists and defines the commands for setting up and managing a trashcan.

Table 1–1: Trashcan Commands

Command	Description
`mktrashcan`	Creates the trashcan
`shtrashcan`	Shows the contents of the trashcan
`rmtrashcan`	Removes the trashcan directory

For example, to attach the trashcan directory `keeper` to the directory `booklist`:

```
# mkdir keeper
# mktrashcan keeper /booklist
   'keeper' attached to '/booklist'
```

To remove a file, and look for it in the trashcan directory:

```
# rm old_titles
# shtrashcan /booklist
   '//keeper' attached to '/booklist'
# cd keeper
# ls
   old_titles
```

To remove the connection between the trashcan and the directory:

```
# rmtrashcan /booklist
   '/booklist' detached
```

1.9 Configuring an AdvFS root File System

There are several advantages to configuring AdvFS for the root file system. You can:

- Restart quickly after a crash. You do not run the `fsck` utility after a crash.
- Use one set of tools to manage all local file systems. All features of AdvFS except the `addvol` and `rmvol` commands are available to manage the root file system, except in a cluster configuration where multivolume root domains are allowed.
- Use AdvFS with LSM to mirror the root file system. This allows your root file system to remain viable even if there is a media failure.

The following restrictions on the AdvFS root file systems are currently enforced:

- Unless you are running a cluster (see *Cluster Administration*), the root domain can contain only one volume. You cannot add volumes to the root domain.
- The volume must start from the beginning of the physical device (`a` or `c` partitions).
- The root fileset must be the first fileset created in the root domain.
- You can assign any name to the root domain and fileset but the same name must be entered in the `/etc/fstab` file.

The root domain should contain only the root fileset. It is not advisable to include the `/usr` and `/var` filesets in the root domain. If you have only one storage device on your system, put other domains on different partitions.

You can put the root file system on an AdvFS volume during the initial base system installation or you can convert your existing root file system after installation. Note that when you install AdvFS as the root file system during the initial installation, root defaults to the `a` partition.

If you construct your own root file system, you must configure it on the `a` or `c` partition. See Section C.2 for instructions on converting an existing UFS root file system to AdvFS. See the *Installation Guide* for instructions on installing AdvFS as the root file system during the initial installation.

1.9.1 Mounting the root File System in Single-User Mode

The root file system is automatically mounted as read-only when the system is booted in single-user mode. You can change the root fileset mount from read-only to read-write by using the `mount -u` command:

```
# mount -u /
```

Use this procedure when you need to make modifications to the root configuration. For example, use it if you need to modify your `/etc/fstab` file.

1.9.2 Changing the Name of the root Domain

You can change the name of the root domain the same as any other domain (see Section 1.6.9). The name of a root domain is stored as the directory name in the `/etc/fdmns` directory and in the entry for root in the `/etc/fstab` file.

1.9.3 Changing the Name of the root Fileset

Changing the name of the root fileset is similar to changing the name of any other fileset (see Section 1.7.9). There are, however, two complications:

- The `renamefset` command requires that the fileset be unmounted, and you cannot unmount the root fileset.
- You must edit the `/etc/fstab` file to change the name of the root fileset. To do this, you must make the root fileset writable. But you cannot mount the root fileset as writable unless the `/etc/fstab` entry is correct.

Therefore, you must use an alternate bootable partition and manipulate the root fileset you are changing as you would an ordinary fileset, make the changes, then reboot the changed fileset as root.

To rename the root fileset:

1. Boot a partition other than the one you want to change. (It can be UFS.)

2. Make a new entry in the /etc/fdmns directory of the booted partition for the fileset whose name you want to change.
3. Change to the new directory and make a symbolic link to the device holding the original fileset.
4. Use the renamefset command to rename the root fileset.
5. Mount the newly named root fileset at a temporary location in order to update its /etc/fstab file.
6. Change the fstab entry to correspond to the new root fileset name.
7. Shut down the alternate system.
8. Reboot the original AdvFS system.

The following procedure changes the name of the root fileset from root_fs to new_root. Assume that the root fileset is in the root_domain domain on /dev/disk/dsk2a.

1. Boot a device other than the one you want to change.
2. Make an entry for tmp_root_domain in the /etc/fdmns directory:

   ```
   # mkdir /etc/fdmns/tmp_root_domain
   ```

3. Change to the new directory and make a symbolic link for tmp_root_domain:

   ```
   # cd /etc/fdmns/tmp_root_domain
   # ln -s /dev/disk/dsk2a
   ```

4. Rename the fileset from root_fs to new_root:

   ```
   # renamefset tmp_root_domain root_fs new_root
   ```

5. Mount the changed root fileset to update the associated /etc/fstab file:

   ```
   # mount tmp_root_domain#new_root /mnt
   ```

6. Edit the /mnt/etc/fstab entry for tmp_root_domain using the vi editor:

   ```
   # cd /mnt/etc
   # vi fstab
   ```

   Locate the following line in the /etc/fstab file:

   ```
   root_domain#root_fs / advfs rw,userquota,groupquota 0 2
   ```

   Edit this line as follows:

   ```
   root_domain#new_root / advfs rw,userquota,groupquota 0 2
   ```

7. Shut down the alternate system.

   ```
   # shutdown -h now
   ```

8. Reboot the AdvFS system.

Note

If you change the root domain and fileset names and do not change the /etc/fstab entries, you cannot boot past single-user mode. Edit /etc/fstab file in single-user mode using an editor to proceed.

1.10 Using AdvFS with LSM

Logical Storage Manager (LSM) allows you to manage all of your storage devices, such as disks, partitions, or RAID sets as a flexible pool of storage from which you create LSM volumes. You can create mirrored and striped LSM volumes and change their attributes as performance and availability needs dictate. For more information see *Logical Storage Manager*.

AdvFS treats LSM volumes just like any other storage. You can use LSM to create and manage the volumes that you use in AdvFS domains.

Structuring domains with LSM storage provides some performance and reliability benefits:

- Data mirroring and RAID allow access to the same information on more than one physical device. This provides backup in the case of a disk failure.
- Mirroring can improve the I/O throughput because files can be accessed from more than one physical location.
- LSM's volume striping is useful when many files are shared. AdvFS stripes individual files.
- Detailed information on disk I/O activity is available with LSM monitoring.
- You can perform backups without shutdown on a mirrored system by detaching the mirror, backing up the detached volume, then reattaching it.

Caution

LSM volumes are not backward compatible. Do not move an LSM volume that you added to a domain on a Version 5 operating system to a system running a Version 4 operating system.

See *Logical Storage Manager* for further information.

1.10.1 Setting Up AdvFS Using LSM Volumes

Use AdvFS commands for AdvFS activities. To manage LSM storage that is part of an AdvFS domain, use LSM commands. If you want to create an LSM volume or place and existing domain under LSM control, see *Logical Storage Manager*.

To use LSM with AdvFS from the command line:

1. Create the LSM volumes with the desired attributes.

   You can specify mirrored volumes, striped volumes, mirrored and striped volumes, volume location, and volume size.

2. Create a domain by using the `mkfdmn` command and identify the LSM volume as the initial volume.
3. If desired, and if you have AdvFS Utilities, create a multivolume domain by using the `addvol` command adding either LSM or non-LSM volumes.

See *Logical Storage Manager* for further information.

Caution

Do not use the `grow` or `shrink` LSM options to change the size of AdvFS domains. Use the `addvol` and `rmvol` commands instead.

You will get an error message if you try to create an AdvFS domain on an LSM volume that is already opened or on a volume that is already labeled as in use by either UFS or AdvFS.

If you already have an AdvFS domain, you can encapsulate the domain into LSM by using the LSM encapsulation tools.

If mirrored or striped LSM volumes are part of an AdvFS domain that also includes non-LSM volumes, you cannot control which files go to the mirrored or striped LSM volumes. To place specific data on mirrored or striped volumes, create an AdvFS domain that contains only LSM volumes that are mirrored or striped. Then, migrate the files you want mirrored or striped in that domain.

The `showfdmn` command output, the LSM Storage Administrator (LSMSA), and the AdvFS GUI (see Appendix D) include LSM storage in the domain information display.

Note

You must use the full LSM pathname when referencing an LSM volume.

The following example creates a domain on an existing LSM volume:

```
# mkfdmn /dev/vol/rootdg/vol01 domain_rs
```

You can also add volumes to an existing domain:

```
# addvol /dev/vol/rootdg/vol02 domain_rs
```

1.10.2 Using LSM for Data Redundancy

You can configure domains with LSM mirrored volumes. This configuration provides failover protection. Because the multiple copies of the volumes contain the same information, either one can be accessed.

See *Logical Storage Manager* for detailed instructions for creating LSM mirrors. Mirrored volumes are added to AdvFS domains with AdvFS commands.

1.10.3 Striping LSM Volumes and Striping AdvFS Files

You can distribute I/O across multiple volumes by striping. AdvFS and LSM striping both work well on a busy system. LSM is better suited for general striping, while AdvFS striping allows the administrator finer-grained control for positioning individual files.

When you choose LSM striping, you configure your LSM volumes for striping before you add them to the domain. With LSM striping all files located on the volumes are striped. You can do this from the command line or from LSMSA (see *Logical Storage Manager*).

You can perform AdvFS striping at any time (see Section 4.13). Therefore, if you find that a file has become large or has heavy I/O requirements, you can use the AdvFS `stripe` command to stripe it.

It is not advisable to do both LSM and AdvFS striping because this might degrade system performance.

2

Managing Quotas

The AdvFS file system allows you to activate quotas to track and control the amount of physical storage that each user, group, or fileset consumes. You must have root user privilege to set and edit quotas. Quota limits never restrict the root user; quota limits only apply to nonroot users.

This chapter covers the following:

- Section 2.1 describes how to use quotas and grace periods.
- Section 2.2 explains user and group quotas.
- Section 2.3 describes fileset quotas.

2.1 Introducing Quotas

The AdvFS quota system is compatible with the quotas of UFS. However, the AdvFS quota system differs in two ways:

- AdvFS differentiates between quota maintenance and quota enforcement. Quota information is always maintained, but you can activate and deactivate enforcement can be activated and deactivated.
- AdvFS supports fileset quotas; that is, you can set quota limits for the filesets in a domain.

The AdvFS file system creates **quota files** to track quotas, grace periods, and fileset usage. Quota files are maintained within the fileset but, unlike UFS, the user cannot delete or create them. Quota files are present in the fileset even if quota limits have not been established.

You can set quota values on the amount of disk storage and on the number of files. In addition, you can set two types of quotas:

- User and group quotas

  AdvFS user and group quotas are similar to UFS quotas. You can set a separate quota for each user or each group of users for each fileset.

- Fileset quotas

  Fileset quotas are unique to AdvFS. You can restrict the space that a fileset itself can use. Fileset quotas are useful when a domain contains multiple filesets. Without fileset quotas, any fileset can use all of the disk space in the domain.

For example, it is useful to set quotas on filesets that contain home directories such as `/usr/users` because these filesets can grow rapidly. Conversely, setting quota limits on the `/tmp` fileset is not recommended because this fileset is likely to fluctuate in size.

Quota files and fileset quotas are saved when the root user specifies a complete backup on a local system.

2.1.1 Quota Limits

Limits are set on disk usage (number of blocks) or on number of files (inodes) or both. Table 2–1 shows the size limits for both types of quota values.

Table 2–1: Quota Size Limits

	User and Group Quotas	Fileset Quotas
Disk Usage	8 billion terabytes*	4 billion terabytes
Number of Files	4 billion	4 billion

* Prior to Version 5.0 of the operating system, the disk usage limit for user and group quotas was 2 terabytes. If your domain was created with an operating system version earlier than Version 5.0, and if you are now running Version 5.0 or later operating system, you must upgrade your domain if you want to take advantage of the new limits (see Section 1.6.3).

All quotas can have a hard limit and soft limit. A **hard limit** cannot be exceeded. No more space can be allocated or files created. A **soft limit** permits a period of time during which the limit can be exceeded as long as the hard limit has not been exceeded.

The root user can set or change hard and soft limits at any time and the changes take effect immediately on filesets where quotas are active. Mounting or unmounting filesets and activating or deactivating quotas have no effect on the limits. Hard and soft limits can be set for users, for groups, and for filesets. The default is no quota limit. You can also:

- Set the hard limit to 0 blocks or files to impose no quota limits
- Set the hard limit to 1 block or file to permit no disk space allocations
- Set the soft limit to 1 block or file and the hard limit to 0 blocks or files to permit disk-space allocations on a temporary basis

Quota limits remain in effect until changed by the root user.

See Section 5.8.2 for directions if your activities cause you to run into limits.

2.1.2 Grace Period

Associated with each soft limit is a **grace period**. The grace period is the amount of time during which the soft limit can be exceeded. This time limit stays in effect until the usage goes below the soft limit quota value.

When the grace period expires, the user or the group cannot create new files or allocate more disk space until enough files are deleted to fall below the soft limit. Updating existing files can cause loss of data.

The timer for the grace period starts when the user exceeds the soft limit. If you allow no grace period, the user can exceed the soft limit only once. The grace period is turned off and reset each time usage drops below the soft limit. If you change the grace period after the user has exceeded the soft limit, the old grace period stays in effect until usage drops below the limit.

You can set grace periods for:

- Users

  For each fileset you can set only one grace period for all users (see Section 2.2.2). If the user grace period expires, the user cannot allocate storage in the fileset until enough files are deleted to fall below the soft limit. The grace periods for the number of blocks and for the number of files do not need to be the same.

- Groups

  For each fileset you can set only one grace period for all groups (see Section 2.2.2). If the group grace period expires, no one in the group can allocate storage in the fileset until enough files are deleted to fall below the soft limit. The group grace period limits all users in the group even if the user grace period is larger.

- Filesets

  The fileset grace period is the same as the group grace period (see Section 2.3.2.2). Any user or group can cause the fileset to exceed its soft limit and thereby evoke the fileset grace period. Any user or group (not necessarily the one that exceeded the limit) can delete files to bring the fileset below the soft limit.

AdvFS sets a default grace period of 7 days. This period can be changed (see Section 2.2.2 and Section 2.3.2). You can:

- Specify the grace period in days, hours, minutes, or seconds
- Set the grace period to 0 days to impose the default grace period of 7 days
- Set the grace period to 1 second to allow no grace period

2.2 User and Group Quotas

The following sections describe quota files and how quotas and grace periods are set for users and groups.

2.2.1 Quota Files

The AdvFS file system keeps user and group quota information in the root directory of the fileset in the `quota.user` and `quota.group` files. These files are created when the fileset is created. They are indexed by user ID and group ID. Each quota file entry contains the following information:

- Hard block limit
- Soft block limit
- Block usage
- Hard file limit
- Soft file limit
- File usage
- Block grace period
- File grace period

Quota files are sparse files (see Section 1.3.3); that is, there are holes in the file where no user IDs or group IDs fall. Use the `ls -l` command to see the space spanned by the `quota.user` or the `quota.group` file. This does not display the actual disk usage. For example:

```
# ls -l quota.user
-rw-r-----  1 root     operator  294912 Jul 20 08:50 quota.user
```

Use the `ls -s` command or the `du -k` command to display how many 1 KB blocks the file actually uses:

```
# ls -s quota.user
16   quota.user

# du -k quota.user
16         quota.user
```

To enforce user or group quotas for a fileset, the `/etc/fstab` file must contain the `userquota` or the `groupquota` mount-point options respectively for the fileset:

```
domain#fileset /mnt advfs rw,userquota,groupquota 0 2
```

If these options are present, you can use the `quotaon` command to start enforcing quotas (see Section 2.2.6).

You can relocate the quota.user and quota.group files to subdirectories of the fileset. However, you cannot relocate them to other filesets nor delete them. If you relocate your files, you must update the /etc/fstab file entry to include the path and name of the relocated file(s) in the userquota and groupquota parameters.

For example, to relocate the quota.user file to the d4 subdirectory and rename it nquot, change the /etc/fstab entry to:

```
dom_1#fset /mnt advfs rw,userquota=/mnt/d4/nquot,groupquota 0 2
```

In this example, the group quota file is not moved.

Whether or not you relocate your quota files, you do not need to enforce quotas. See Section 2.2.7 for instruction on deactivating user and group quotas.

2.2.2 Setting Quotas and Grace Periods

You are not required to set both user and group quotas. If you specify a group quota, it applies to all users belonging to that group. If you specify a user quota that is larger than the group quota, it has no effect because the group quota takes effect before the user quota is reached.

Use the edquota command to set the quotas for users and groups and use it again to change the grace period. You must mount the fileset before you can set quotas and grace periods. Note that for each fileset, only one grace period can be set for all users and one grace period can be set for all groups. However, you do not have to set the same grace period for the number of blocks and for the number of files. The group grace period you set is also entered as the fileset grace period (see Section 2.3.2.2).

Follow these general steps to set user and group quotas and grace periods:

1. Add quota file mount-point options (userquota, groupquota) to the /etc/fstab file.
2. Enter the edquota command with the -u option to set user quotas or the -g option to set group quotas. The editor specified by the EDITOR environment variable displays an ASCII representation of the quota file. If the EDITOR environment variable is not set, the vi editor is used.
3. When the user or group quota information is displayed, modify the values in the limits fields as needed. Then, exit the editor, saving the changes.
4. To set user or group grace periods, enter the edquota command with the -ut option for user or the -gt option for group. Note that the fileset grace period is the same as the group grace period.

5. When the grace period information is displayed, modify the grace period as desired. Then, exit the editor, saving the changes.

To change the default editor and for more information, see `edquota`(8).

The quota limits you set for a fileset do not take effect until you activate them by using the `quotaon` command (see Section 2.2.5 and Section 2.2.6). If you are changing quota limits and quotas have already been activated for a fileset, the new limits become effective immediately.

If you have set quotas for a single user, you can use the `edquota -p` command, which allows that user's entry to serve as a prototype to be applied to other users that you specify (see Section 2.2.3).

2.2.2.1 Setting User Quotas

The following procedure sets quotas for the user `user5`:

1. If they do not already exist, add quota mount-point options to the `/etc/fstab` file. Note that there can be no spaces in the list of options delimited by commas; that is, from `rw` through `groupquota`:

   ```
   domain_1#eb1  /eb1  advfs rw,userquota,groupquota 0 2
   domain_2#eb3  /eb3  advfs rw,userquota,groupquota 0 2
   domain_4#eb4  /eb4  advfs rw,userquota,groupquota 0 2
   ```

2. Enter the `edquota -u` command followed by the user name. If you specify more than one user name, the edits affect all users named. The command creates a temporary file with an ASCII representation of the current quotas assigned to the named users and invokes an editor to allow you to modify the file:

   ```
   # edquota -u user5
    Quotas for user user5:
    /eb1: blocks in use: 0, limits (soft = 0, hard = 0)
            inodes in use: 0, limits (soft = 0, hard = 0)
    /eb3: blocks in use: 0, limits (soft = 0, hard = 0)
            inodes in use: 0, limits (soft = 0, hard = 0)
    /eb4: blocks in use: 0, limits (soft = 0, hard = 0)
            inodes in use: 0, limits (soft = 0, hard = 0)
   ```

   The values for `blocks in use` and `inodes in use` are the number of 1 KB blocks and the number of files currently in use for each fileset. You cannot change them. Soft and hard limits of 0 (zero) indicate that no limits have been set.

3. To change user quotas for `user5` for fileset `eb3`, edit the `quota.user` file. Enter the new limits for disk usage on the `blocks` line and enter the new limits for the number of files on the `inodes` line:

   ```
    /eb3: blocks in use: 0, limits(soft=5000, hard=10000)
            inodes in use: 0  limits(soft= 100, hard=  200)
   ```

4. Exit the editor, saving the changes.

If quotas have already been activated for fileset eb3, the new limits become effective immediately. If quotas are not yet activated for the fileset, the limits become effective as soon as quotas are activated (see Section 2.2.6).

Quotas set for the group to which the user belongs take precedence over quotas set for the user of the fileset.

2.2.2.2 Setting the User Grace Period

When you impose soft limits for a fileset, you can set a grace period for that fileset. If you do not specify a grace period, the grace period remains at the AdvFS default of 7 days. You can set different grace periods for the number of blocks and for the number of files. If the group grace period is less than a user grace period, the user is limited by the group grace period.

The following procedure sets the grace period for all users of fileset eb3:

1. Run the edquota -ut command. The command creates a temporary file with an ASCII representation of the current grace period and invokes an editor to allow you to modify the file.

   ```
   # edquota -ut
   Time units may be: days, hours, minutes, or seconds
   Grace period before enforcing soft limits for users:
   /eb1: block grace period: 7 days, file grace period: 7 days
   /eb3: block grace period: 7 days, file grace period: 7 days
   /eb4: block grace period: 7 days, file grace period: 7 days
   ```

2. To set the user grace period for the number of blocks and for the number of files for eb3, edit the file to change the existing grace period:

   ```
   Time units may be: days, hours, minutes, or seconds
   Grace period before enforcing soft limits for users:
   /eb1: block grace period: 7 days, file grace period:7 days
   /eb3: block grace period: 2 days, file grace period:3 days
   /eb4: block grace period: 7 days, file grace period:7 days
   ```

3. Exit the editor, saving the changes.

If quotas have already been activated for the fileset eb3, the new grace period becomes effective immediately. If a user has already exceeded the soft limit, the grace period becomes effective when the usage drops below the soft limit. If quotas are not yet activated for the fileset, the grace period becomes effective as soon as quotas are activated (see Section 2.2.6).

2.2.2.3 Setting Group Quotas

The following procedure sets quotas on eb3 for the group rsgusers:

1. If they do not already exist, add quota mount-point options to the /etc/fstab file. Note that there can be no spaces in the list of options delimited by commas; that is, from rw through groupquota:

   ```
   domain_1#eb1  /eb1  advfs rw,userquota,groupquota 0 2
   domain_2#eb3  /eb3  advfs rw,userquota,groupquota 0 2
   domain_4#eb4  /eb4  advfs rw,userquota,groupquota 0 2
   ```

2. Enter the edquota -g command. If you specify more than one group name, the edits affect all groups named. The command creates a temporary file with an ASCII representation of the current quotas assigned to the named groups and invokes an editor to allow you to modify the file.

   ```
   # edquota -g rsgusers
   Quotas for group rsgusers:
   /eb1: blocks in use: 0, limits (soft=0, hard=0)
           inodes in use: 0, limits (soft=0, hard=0)
   /eb3: blocks in use: 0, limits (soft=0, hard=0)
           inodes in use: 0, limits (soft=0, hard=0)
   /eb4: blocks in use: 0, limits (soft=0, hard=0)
           inodes in use: 0, limits (soft=0, hard=0)
   ```

   The values for blocks in use and inodes in use are the current number of 1 KB blocks in use and the number of files for each fileset. You cannot change them. Soft and hard limits of 0 (zero) indicate that no limits have been set.

3. To change the group quotas for eb3, edit the group.quota file. Enter the new limits for disk usage on the blocks line and enter the new limits for the number of files on the inodes line:

   ```
    /eb3: blocks in use:0, limits(soft=60000, hard=80000)
            inodes in use:0, limits(soft= 6000, hard= 8000)
   ```

4. Exit the editor, saving the changes.

If quotas have already been activated for fileset eb3, these limits become effective immediately. If quotas are not yet activated for eb3, these limits become effective as soon as quotas are activated (see Section 2.2.6). Quotas set for the group take precedence over quotas set for the individual.

2.2.2.4 Setting the Group Grace Period

When you impose soft limits, you can set one grace period per fileset for all groups. If you do not specify a grace period, the grace period remains the AdvFS default of 7 days. You can set different grace periods for the number of blocks and for the number of files. The group grace period takes precedence over all user grace periods.

The following procedure sets the grace period for all groups for fileset eb3:

1. Run the edquota -gt command. The command creates a temporary file with an ASCII representation of the current grace period and invokes an editor to allow you to modify the file.

```
# edquota -gt
Time units may be: days, hours, minutes, or seconds
Grace period before enforcing soft limits for groups:
/eb1: block grace period: 7 days, file grace period:7 days
/eb3: block grace period: 7 days, file grace period:7 days
/eb4: block grace period: 7 days, file grace period:7 days
```

2. To set the group grace period for the number of blocks and for the number of files for eb3, edit the file to change the existing grace period:

```
Time units may be: days, hours, minutes, or seconds
Grace period before enforcing soft limits for groups:
/eb1: block grace period: 7 days, file grace period:7 days
/eb3: block grace period:12hours, file grace period:5 days
/eb4: block grace period: 7 days, file grace period:7 days
```

3. Exit the editor, saving the changes.

If quotas have already been activated, this grace period becomes effective immediately unless a group has already exceeded the soft limit for eb3. In that case, the new grace period becomes effective for that group when the group usage drops below the soft limit. If quotas are not yet activated, the group grace period becomes effective as soon as they are activated (see Section 2.2.6).

2.2.3 Setting Quotas for Multiple Users and Groups

The AdvFS file system allows you to use a single command to modify quotas for a list of users or groups so you do not need to access and enter values for each one individually. Note that you do not have to set multiple grace periods because, for each user or group quota, one grace period per fileset applies to all users or to all groups.

Use the edquota -p command to take existing quota information and establish it as a prototype user or group quota. Then apply the prototype to one or more users or groups.

For example, to set up all student accounts to have the same disk usage quota:

1. Use the edquota command with the desired limits to establish a set of quotas for a single student.

2. Use the `edquota -p` command to apply the quotas set up for the first user to other student accounts.

2.2.3.1 Setting Quotas for Multiple Users

The following procedure sets up prototype-user quotas that are then used to modify the quotas for other users:

1. Set quotas for one user, `user5` (see Section 2.2.2.1):

```
# edquota -u user5
Quotas for user user5:
/eb1:blocks in use:0, limits(soft= 20000,hard= 30000)
       inodes in use:0, limits(soft=   350,hard=   500)
/eb3:blocks in use:1 ,limits(soft= 30000,hard= 40000)
       inodes in use:4, limits(soft=   400,hard=   550)
/eb4:blocks in use:2, limits(soft= 10000,hard= 20000)
       inodes in use:1, limits(soft=   150,hard=   200)
/eb5:blocks in use:2, limits(soft=100000,hard=150000)
       inodes in use:1, limits(soft=  5000,hard=  7000)
```

2. To create quotas for new users `user7`, `user8`, and `user9`, use the quotas from user `user5` as a prototype:

```
# edquota -p user5 -u user7 user8 user9
```

3. To verify that the quotas were set, run the `edquota` command for `user7`:

```
# edquota -u user7
Quotas for user user7:
/eb1:blocks in use:0, limits(soft= 20000,hard= 30000)
       inodes in use:0, limits(soft=   350,hard=   500)
/eb3:blocks in use:0, limits(soft= 30000,hard= 40000)
       inodes in use:0, limits(soft=   400,hard=   550)
/eb4:blocks in use:0, limits(soft= 10000,hard= 20000)
       inodes in use:0, limits(soft=   150,hard=   200)
/eb5:blocks in use:0, limits(soft=100000,hard=150000)
       inodes in use:0, limits(soft=  5000,hard=  7000)
```

2.2.3.2 Setting Quotas for Multiple Groups

The following procedure sets up prototype group quotas that are then used to modify the quotas for another group:

1. Set quotas for the group `rsgusers` (see Section 2.2.2.3):

```
# edquota -g rsgusers
Quotas for group rsgusers:
/eb1:blocks in use:0, limits(soft=100000,hard=200000)
       inodes in use:0, limits(soft= 10000,hard= 20000)
/eb3:blocks in use:0, limits(soft=300000,hard=400000)
       inodes in use:0, limits(soft= 30000,hard= 40000)
```

```
/eb4:blocks in use:0, limits(soft=500000,hard=600000)
       inodes in use:0, limits(soft= 50000,hard= 60000)
/eb5:blocks in use:0, limits(soft=350000,hard=450000)
       inodes in use:0, limits(soft= 35000,hard= 45000)
```

2. To create quotas for a new group, `rsgstudents`, use the quotas from group `rsgusers` as a prototype:

   ```
   # edquota -p rsgusers -g rsgstudents
   ```

3. To verify that the quotas were set, run the `edquota` command for `rsgstudents`:

   ```
   # edquota -g rsgstudents
   Quotas for group rsgstudents:
   /eb1:blocks in use:0, limits(soft=100000,hard=200000)
          inodes in use:0, limits(soft= 10000,hard= 20000)
   /eb3:blocks in use:0, limits(soft=300000,hard=400000)
          inodes in use:0, limits(soft= 30000,hard= 40000)
   /eb4:blocks in use:0, limits(soft=500000,hard=600000)
          inodes in use:0, limits(soft= 50000,hard= 60000)
   /eb5:blocks in use:0, limits(soft=350000,hard=450000)
          inodes in use:0, limits(soft= 35000,hard= 45000)
   ```

2.2.4 Verifying File and Disk Space Usage

Even if you are not enforcing quotas, you can monitor file and disk space usage by using the `df`, `showfdmn`, `vdf`, and `showfsets` commands. Use the `-k` option to display information in kilobytes. See the appropriate reference pages for more information.

If you are enforcing quotas, you can periodically verify your quota setup. You can display user and group quota information in a number of ways. If you are not the root user, you can display information only for your own files. The root user can display all user and all group quota information for all filesets. All disk quota values are displayed in 1 KB blocks.

The commands shown in Table 2–2 are useful for examining disk space and file usage for filesets for which user and group quotas are enforced.

Table 2–2: User and Group Usage Commands

Command	Description
`ncheck`	Prints the tag and full pathname for each file in the fileset
`quot`	Summarizes fileset ownership
`quota`	Displays disk usage and limits by user or group
`quotacheck`	Checks fileset quota consistency
`repquota`	Summarizes quotas for a fileset

See the command reference pages for more information.

If your disk usage information appears to be corrupt, see Section 5.1 for instructions on how to correct this.

2.2.4.1 Printing the Tag and Full Pathname for Each File

The ncheck command lists files by **tag** (equivalent to inode) number. By piping the output to the sort command, you can use the sorted output as input for the quot command to list all files and their owners. Use the following format to generate the listing:

```
ncheck domain#fileset |sort +0n| quot -n domain#fileset
```

2.2.4.2 Summarizing Fileset Ownership

The quot command displays block usage and the number of files in the fileset that each user owns. If you do not specify a fileset, the command processes all filesets in the /etc/fstab file that include the ro, rw, and rq mount options.

The quot command entered with no options displays only blocks:

```
# quot domain_1#set_1
domain_1#set_1:
34128    root
  816    user5
```

The quot -f command displays both blocks and files:

```
# quot -f domain_1#set_1
domain_1#set_1:
34128     125   root
  816       9   user5
```

2.2.4.3 Displaying Disk Usage and Limits

The quota command displays the block usage, the number of files, and the quotas for a user or group. Users can run this command to look at their own disk space usage. The root user can look at usage for the whole system.

You can choose to display quota information for users or groups, for all filesets with usage over quota, or for all mounted filesets regardless of whether quotas are activated.

The quota command displays the block usage of the fileset, soft limit (quota), hard limit (limit), grace period, and the number of files used for each user. An asterisk (*) in a column means that a soft quota limit has been exceeded. Note that the grace period is not displayed unless the soft limit has been exceeded. Use the edquota command to view complete quota specifications.

The following example displays quota information for the user user5:

```
# quota -u user5
Disk quotas for user user5 (uid 446):
Filesystem   blocks  quota   limit grace files quota limit  grace
         /       60    100     150           3    10    20
      /usr     5071*  5000   10000 24:40     2    20    40
    /eb1      816  20000   30000           9   350   500
    /eb2    22032  50000  200000           2  2000  4000
    /eb3     2344  10000   15000         370  1000  2000
    /eb4    18023* 10000   20000 7days     3   100   150
    /eb5    32012* 20000   50000 7days     0  2000  3000
```

The following example displays quota information for the group rsgusers:

```
# quota -g rsgusers
Disk quotas for group rsgusers (gid 15):
Filesystem blocks  quota   limit grace files quota limit  grace
         /    118    200     300           2    20    40
      /usr  13184* 10000   20000 7days     2    40    80
    /eb1  36136 100000  200000         124 10000 20000
    /eb2  44064 200000  400000           4  2000  4000
    /eb3   3587  30000   60000         628  3000  5000
    /eb4  51071 150000  300000           6  1050  1800
    /eb5  61044 100000  200000           3 10000 20000
```

2.2.4.4 Verifying Quota Consistency

The quotacheck command verifies that the actual blocks used and the number of files are consistent with the established limits. It examines user and group files, builds a table of current disk usage, and compares this table with that stored in the disk quota file.

The quotacheck command requires that filesets be mounted. Use the -v option (verbose) to display inconsistencies found and procedures performed during the checking process.

By default both user and group quotas are checked, but you can specify either by using the -u option for user or the -g option for group.

The following example displays a verbose check of the fileset set_1 that displays no inconsistencies:

```
# quotacheck -v domain_1#set_1
 *** Checking user and group quotas for domain_1#set_1 (/eb1)
```

The following example checks all filesets that have quotas defined in the /etc/fstab file. In this example the quotacheck command fixes inconsistencies in the /usr directory:

```
# quotacheck -va
*** Checking user and group quotas for /dev/rdisk/dsk0g (/usr)
*** Checking user and group quotas for domain_1#set_1 (/eb1)
/usr: root   fixed: inodes 3057 -> 3022 blocks 100616 -> 123440
/usr: system fixed: inodes 2483 -> 2488 blocks 91721  -> 114568
/usr: adm    fixed: inodes 280  ->  240 blocks   487  ->    464
```

In this display, `inodes` is the number of files and `blocks` is the block usage.

2.2.4.5 Summarizing Quotas by Fileset

The `repquota` command displays the actual disk usage and quotas for the specified filesets. The fileset must have a quota entry in the `/etc/fstab` file to be included in the summary. By default user quotas are reported, but you can specify which quotas are reported by using the `-u` option for user or the `-g` option for group. If you specify the `-a` option, both the user and group quotas are displayed.

For each user or group, the `repquota` command prints the current number of files, the amount of space used, and the quota limits established by the `edquota` command.

The following example summarizes quotas for a single fileset mounted on `/eb1`:

```
# repquota -v /eb1
*** Report for user quotas on /eb1 (domain_1#set_1)
                    Block limits              File limits
User       used    soft    hard  grace  used  soft  hard  grace
root  --  34088       0       0          123     0     0
user5 --    816   20000   30000            9   350   500
```

The following example displays user and group quota information for all filesets in the `/etc/fstab` file that have quotas defined. Note that this example contains both UFS and AdvFS files:

```
# repquota -va
*** Report for group quotas on /usr (/dev/disk/dsk0g)
                    Block limits              File limits
Group         used  soft  hard  grace   used  soft  hard  grace
system   -- 114568     0     0          2488     0     0
daemon   --    144     0     0             1     0     0
uucp     --    801     0     0             8     0     0
mem      --   1096     0     0            10     0     0
bin      -- 108989     0     0          3219     0     0
mail     --    209     0     0             2     0     0
terminal --     56     0     0             2     0     0
adm      --    464     0     0           240     0     0
operator --    392     0     0             3     0     0
211      --   6937     0     0            33     0     0
*** Report for user quotas on /usr (/dev/disk/dsk0g)
```

```
                     Block limits              File limits
User          used  soft  hard  grace   used  soft  hard  grace
root     -- 123440     0     0          3022     0     0
bi       -- 102534     0     0          2940     0     0
uucp     --    729     0     0             7     0     0
adm      --      1     0     0             1     0     0
user5    --     15    18    24             1     0     0
kraetsch --   6937     0     0            35     0     0
*** Report for group quotas on /eb1 (domain_1#set_1)
                     Block limits              File limits
Group         used  soft  hard  grace   used  soft  hard  grace
system   --  22816     0     0            50     0     0
daemon   --  12088     0     0            82     0     0
*** Report for user quotas on /eb1 (domain_1#set_1)
                     Block limits              File limits
User          used  soft  hard  grace   used  soft  hard  grace
root     --  34088     0     0           123     0     0
user5    --    816 20000 30000             9   350   500
*** Report for group quotas on /eb3 (domain_2#set_1)
                     Block limits              File limits
Group        used   soft  hard  grace   used  soft  hard  grace
system   --  1593      0     0             6     0     0
*** Report for user quotas on /eb3 (domain_2#set_1)
                     Block limits              File limits
User         used   soft  hard  grace   used  soft  hard  grace
root     --  1593      0     0             6     0     0
```

2.2.5 Activating Quotas at System Start-Up

You can automatically start user and group quota enforcement during system initialization.

1. Edit the /etc/fstab file entry to change the fourth field to rw,userquota,groupquota.
2. Use the rcmgr command to add the QUOTA_CONFIG option to the /etc/rc.config.common file:

   QUOTA_CONFIG=**"yes"**

   This entry causes the /sbin/init.d quota script to run the quotaon command for file systems where the userquota or groupquota options are specified in the /etc/fstab file. Quota enforcement is activated for the mounted fileset the next time and every time you reboot.

Note

If you unmount a fileset when quota enforcement is active, you must explicitly reactivate quota enforcement by using the quotaon command when you remount the fileset. This must be done even if there is a QUOTA_CONFIG=”yes” entry in the /etc/rc.config.common.

Setting the QUOTA_CONFIG option to “yes” also causes the quotacheck command, which checks file system quota consistency, to run for UFS file systems that have quota options specified in the /etc/fstab file. By default, the quotacheck command is not run for AdvFS file systems because the AdvFS metadata transaction logging keeps storage allocations and quotas synchronized.

To run the quotacheck command at startup for AdvFS file systems with quota options in the /etc/fstab file, use the rcmgr command to add the following option to the /etc/rc.config.common file:

```
/usr/sbin/rcmgr -c set QUOTACHECK_CONFIG -a
```

To restore the default UFS-only behavior for the quotacheck command, change to the following:

```
/usr/sbin/rcmgr -c set QUOTACHECK_CONFIG ""
```

2.2.6 Activating Quotas Manually

If your system is running, enter the quotaon command to activate new quotas for a mounted fileset. To establish new user or group quotas do the following:

1. Edit the /etc/fstab file entry for your fileset to add userquota and groupquota to the mount point.
2. Run the edquota command to enter the hard and soft limits and to enter the grace period.
3. Run the quotaon command to activate the new quotas.

If your system is set up to initialize quotas at system startup (see Section 2.2.5), you do not need to run the quotaon command again unless you have unmounted your fileset. If your system is set up without quota enforcement (see Section 2.2.7), you must run the quotaon command to start enforcement each time you reboot.

The following example activates quotas for the filesets for which quota values were set in the previous sections:

```
# quotaon -av
/eb1:  group quotas turned on
/eb1:  user quotas turned on
/eb3:  group quotas turned on
/eb3:  user quotas turned on
/eb4:  group quotas turned on
```

```
/eb4:  user quotas turned on
```

By default, both user and group quotas are affected by the quotaon and quotaoff commands. You can choose to activate quotas either for users (with the -u option) or for groups (with the -g option). You can also specify the filesets for which user or group quotas are enforced (see Section 2.3.5 for information on fileset quotas).

2.2.7 Deactivating Quotas

You can turn off quota enforcement either temporarily or permanently. You can obtain file and disk space usage information regardless of whether you are enforcing quotas.

The quotaoff command turns off quota enforcement until the quotaon command is run again either manually or through system initialization that turns quotas on.

The umount command turns off quotas before it unmounts a fileset. If you remount the fileset, you must run the quotaon command to enforce user and group quotas for the fileset.

If you want to permanently turn quotas off for a user or group, use the edquota command to set quota limits to 0 (zero). To prevent quotas from ever being activated for a fileset, run the quotaoff command. Then, remove the userquota and groupquota entries for the fileset in the /etc/fstab file.

2.3 Fileset Quotas

The following sections describe fileset quota files and how quotas and grace periods are set for filesets from the command line. For information on setting fileset quotas from the AdvFS GUI, see Appendix D.

2.3.1 Quota Files

AdvFS keeps fileset soft and hard limits in the structural information associated with the fileset. You do not have direct access to this file. It contains the same type of information that the user and group quota files contain: hard and soft limits for the number of blocks and hard and soft limits for the number of files. For a given fileset, the fileset grace period is the same as the group grace period.

2.3.2 Setting Quotas and Grace Periods

Fileset quotas limit the number of files or the amount of disk space a fileset can use. You can set both soft and hard limits. If fileset quotas are not imposed, any fileset has access to all of the available disk space in the domain. The fileset quotas are set by using the chfsets command. If fileset quotas are set, they are activated whenever you mount the fileset.

The fileset grace period is the same as the group grace period (see Section 2.3.2.2). If you do not set a grace period, the grace period remains at the AdvFS default grace period of 7 days.

Use the chfsets command to define fileset quota values. You can use the following options:

-F	Sets a soft limit for the number of files
-f	Sets a hard limit for the number of files
-B	Sets a soft limit for block usage
-b	Sets a hard limit for block usage

The command displays both the old and new limits.

2.3.2.1 Setting Fileset Quotas

The following procedure sets fileset quotas for the set_1 fileset in the dmn_2 domain. Note that unlike the quota commands, the showfsets command displays block usage in 512-byte blocks. If you want to display kilobyte values, use the -k option.

1. To display existing fileset quotas, use the showfsets command:

   ```
   # showfsets dmn_2 set_1
   set_1
      Id           : 2feff762.00034e3f.1.8001
      Clone is     : set_1_clone
      Files        :      7,  SLim=      0,  HLim=        0
      Blocks (512) :    118,  SLim=      0,  HLim=        0
      Quota Status : user=on  group=on
      Object Safety: off
      Fragging     : on
      DMAPI        : off
   ```

   Here SLim is the soft limit and HLim is the hard limit for the number of files and the current block usage.

2. Use the chfsets command to set the quotas. Note that the arguments for block usage for the chfsets command are in units of 1 KB, not 512 bytes as displayed by the showfsets command display.

   ```
   # chfsets -F 10000 -f 20000 -B 250000 -b 500000 dmn_2 set_1
   ```

   Here -F sets the soft limit for the number of files, -f sets the hard limit for the number of files, -B sets the soft limit for block usage, and -b sets the hard limit for block usage.

3. To verify the new fileset quotas, rerun the showfsets command:

   ```
   # showfsets dmn_2 set_1
   set_1
      Id           : 2feff762.00034e3f.1.8001
      Clone is     : set_1_clone
      Files        :      7,  SLim=  10000,  HLim=    20000
      Blocks (512) :    118,  SLim= 500000,  HLim=  1000000
      Quota Status : user=on  group=on
      Object Safety: off
      Fragging     : on
      DMAPI        : off
   ```

   Note that the soft limit for the number of blocks is 500000, twice the number 250000 that was input by using the chfsets command; similarly, the hard limit for the number of blocks appears to double from 500000 to 1000000. To avoid this confusion, enter the showfsets command with the -k option to display blocks in 1 KB units.

2.3.2.2 Setting the Grace Period

The fileset grace period is the same as the group grace period and cannot be modified independently (see Section 2.2.2). Therefore, if you use the edquota -gt command to change the grace period for which a fileset can exceed its soft limits, you also change the group grace period and vice versa. The default AdvFS grace period of 7 days remains in effect until you change it.

You can set only one grace period per fileset, but you can set different grace period values for block usage and number of files. The grace period applies to all users and all groups. If the grace period is reset, the new grace period for the fileset takes effect immediately unless the fileset has already exceeded its soft limits. In that case, the new grace period becomes effective once the fileset drops below the soft limit.

The following procedure sets the grace period for the filesets eb1, eb3, and eb4:

1. Run the edquota command with the group grace period options. The command creates a temporary file with an ASCII representation of the current grace period and invokes an editor to allow you to modify the file:

   ```
   # edquota -gt
   Time units may be: days, hours, minutes, or seconds
   Grace period before enforcing soft limits for groups:
   /eb1: block grace period: 7 days, file grace period:7 days
   /eb3: block grace period: 7 days, file grace period:7 days
   /eb4: block grace period: 7 days, file grace period:7 days
   ```

2. To change the fileset grace period for the number of blocks and for the number of files, edit the file to change the existing grace period:

```
Time units may be: days, hours, minutes, or seconds
Grace period before enforcing soft limits for groups:
/eb1: block grace period: 7 days, file grace period:7 days
/eb3: block grace period:12hours, file grace period:5 days
/eb4: block grace period: 7 days, file grace period:7 days
```

3. Exit the editor, saving the changes.
4. Optionally, run the `showfsets -q` command to display the time (if exceeded) and fileset usage and limits:

```
# showfsets -q eb_domain
               Block (512) Limits  File Limits
Fileset  BF  used soft hard grace  used soft hard grace
fileset1 +-  1750 1500 2000 11:32    35  300  400
```

In this example, the plus sign (+) in the `BF` field means that the soft limit for block usage is exceeded. An asterisk (*) indicates that the hard limit has been reached.

2.3.3 Setting Quotas for Multiple Filesets

You can set quota limits for multiple filesets in a domain by listing more than one fileset name when you run the `chfsets` command (see Section 2.3.2).

For example, to change the hard limits for the `data` and `data2` filesets in `eb1_domain`, enter the names of both filesets after the `chfsets` command:

```
# chfsets -b 1000 -f 200 eb1_domain data data2
```

Setting fileset quotas automatically activates the quotas.

2.3.4 Verifying File and Disk Space Usage

To examine how system resources are being used, look at fileset activity. The commands described in the following sections are useful for examining disk space and the file usage of filesets. The command reference pages contain additional information.

2.3.4.1 Displaying Fileset Space

The `df` command displays the available disk space and the disk space used for a fileset. The command calculates capacity using the lower (hard or soft) limit for the amount of space available:

- If a fileset quota has been set, the command displays the amount of space remaining until the quota limit is reached.
- When both soft and hard quota limits are set, the command calculates the disk space available using the lower limit.

- If there is less space in the domain than is allowed by the fileset quota, the command displays the actual space available in the domain.
- If fileset quotas have not been established, the command displays the available domain size; all unused space is available to each fileset.

The following example displays the amount of space available for `fileset_1`:

```
# df /fileset_1
Filesystem        512-blocks  Used Avail  Capacity  Mounted on
eb_domain#fileset_1   1500  1750     0      117%  /fileset_1
```

Because the usage is over the limit, the capacity is determined by the actual space used (1750/1500) and appears as more than 100%. If usage is not over the limit, capacity is calculated as (used)/(used + available).

AdvFS calculates each fileset capacity independently. If the domain has multiple filesets, all unused space is available for each fileset unless fileset quotas limit the space. As a result, the total capacity of all filesets in the domain can appear to be greater than 100%. In the following example, the filesets `domain_1#eb3` and `domain_1#eb4` each can use all of the available disk space from the volumes in `domain_1`:

```
# df
  Filesystem   512-blocks    Used    Avail Capacity  Mounted on
domain_1#eb3    2000000  390820    98864    80%    /eb3
domain_1#eb4    2000000  271580    98864    73%    /eb4
```

2.3.4.2 Displaying Fileset Limits

The `showfsets -q` command displays file usage, hard and soft limits, and grace period information for the filesets in the specified domain. It display s the block usage, the block usage limit, the number of files, and the file limit. The correct information is displayed only if the fileset is mounted.

The following example displays fileset information for the domain `eb_domain`:

```
# showfsets -q eb_domain
         Block (512) Limits      File Limits
Fileset  BF  used soft hard grace  used soft hard grace
fileset1 +-  1750 1500 2000 11:32    35  300  400
```

In this example, the plus sign (+) in the `BF` field means that the soft limit for block usage is exceeded. An asterisk (*) would indicate that the hard limit has been reached.

2.3.4.3 Displaying Domain Space

The showfdmn command is useful for obtaining domain statistics that you can use to make decisions about filesets and their quotas. The command displays the attributes of a domain and information about each volume in the domain. For single-volume or multivolume domains, the command displays the total volume size, the total number of free blocks, and the total percentage of volume space currently allocated.

```
# showfdmn usr_domain
               Id         Date Created LogPgs Version Domain Name
39cbf2d6.0002691e Sat Jan 20 17:01 2000   512       4 usr_domain

 Vol  512-Blks    Free % Used Cmode Rblks Wblks Vol Name
  1L  10605520 7757728    27%    on   256   256 /dev/disk/dsk8d
```

2.3.4.4 Displaying Domain and Fileset Space

The /sbin/advfs/vdf utility reformats output from the showfdmn, showfsets, and df commands to display information about the disk usage of AdvFS domains and filesets. It clarifies the relationship between a domain's disk usage and its fileset's disk usage.

The utility is subject to the following restrictions:

- All filesets must be mounted.
- The disk space used by AdvFS fileset clones is not calculated.
- The command does not produce valid results for filesets that are NFS mounted.

You can specify either a domain or a fileset name for the vdf command. If you specify only a fileset name, the output is the same as that of the df command. If you specify a domain, the utility also displays the number of disk blocks used for metadata. If you specify the -l option with either a domain or a fileset, both domain and fileset information is displayed. The domain metadata displayed is the total metadata shared by all filesets in the domain.

The following example displays the summary information for the domain usr_domain:

```
# /sbin/advfs/vdf usr_domain
Domain     512-blocks   Metadata    Used   Available  Capacity
usr_domain      65536      11219   47549        6768       89%
```

The following example examines the domain eba that contains two filesets. Each has a quota limit of 60,000 blocks.

```
# /sbin/advfs/vdf -l eba
Domain     512-blocks   Metadata    Used   Available  Capacity
```

```
eba           266240       5824   29128      231288       13%

Fileset    QuotaLimit              Used   Available  Capacity
ebfsa          60000              20800       39200       35%
ebfs2a         60000               8328       51672       14%
```

The following example displays the domain ebb that contains two filesets with no quota limits. In this case, the total space available to each fileset is the same as the domain total.

```
# /sbin/advfs/vdf -l ebb
Domain     512-blocks   Metadata    Used   Available  Capacity
ebb           266240       5824   29128      231288       13%

Fileset    QuotaLimit              Used   Available  Capacity
ebfsb              -              20800      231288       35%
ebfs2b             -               8328      231288       14%
```

2.3.5 Activating Quotas

Running the command automatically activates fileset quotas immediately (see Section 2.3.3). No further steps are needed. Fileset quotas are in effect whenever you mount the fileset.

2.3.6 Deactivating Quotas

You can turn off quota enforcement either temporarily or permanently by running the chfsets command with the hard and soft limits set to 0 (zero) to deactivate quotas on a fileset. You can obtain file and disk space usage information regardless of whether you are enforcing quotas.

3

Backing Up and Restoring Data

The AdvFS file system provides extended file system backup capabilities with the vdump and vrestore commands. The rvdump and rvrestore commands provide the same capabilities for remote storage devices. In addition, you can use the AdvFS clonefset utility with the vdump command to back up mounted AdvFS filesets. AdvFS fileset clones provide a static snapshot of files for the backup.

This chapter covers the following:

- Section 3.1 introduces the vdump and vrestore commands.
- Section 3.2 explains ways to save data.
- Section 3.3 discusses clones.
- Section 3.4 explains database backup.
- Section 3.5 details methods to restore data.
- Section 3.6 discusses AdvFS and NetWorker.

3.1 Overview

You can use the vdump command to back up AdvFS filesets as well as UFS and other standard file systems. This means that you can have a single backup utility for your facility. The dump and restore commands function differently from the vdump and vrestore commands. The dump command works at the inode level so it can handle only UFS files. The vdump command works at the file level. It scans the directories and uses regular POSIX file system calls to access directories and files. See vdump(8) and vrestore(8) for more information.

In this section only the vdump and vrestore commands are discussed, but the rvdump and rvrestore commands can be substituted for remote operation.

Note

You do not have to be root user to use the vdump and vrestore commands. However, you must have write permission for the directory to which you want to restore files. Only the root user can save and restore the AdvFS quota files and the fileset quotas.

Caution

The vrestore command for operating system versions earlier than Version 4.0 does not properly restore information saved by the vdump command for Version 4.0 or later. The vdump and vrestore dump file formats are compatible in Version 4 and Version 5 operating systems.

3.2 Backing Up Data

The vdump command creates a list of fixed-size blocks, called a **saveset**, as it copies all files that are new or have changed after a certain date to the default storage device or the device that you specify.

A saveset can span multiple tapes or a tape can contain multiple savesets. Savesets on tapes are delimited by file marks that are written when the vdump command closes the saveset.

The vdump command has the following form:

```
vdump options mount_point
```

For example, to dump the filesets mounted at /psm to tape:

```
# vdump -0 -f /dev/tape/tape0_d1 /psm
```

3.2.1 Unique Features of the vdump Command

The vdump command has a number of functions that the UFS dump command does not have. You can:

- Save mounted filesets.
- Choose a subdirectory that you want to back up. You do not need to dump an entire fileset.
- Compress files to minimize the saveset size.
- Specify the number of in-memory buffers. You can maximize throughput by choosing a number compatible with your storage device.
- Display the current vdump version number.
- Display help information during the dump process.
- Limit your display to error messages. You do not need to display warning messages.

- Display the names of files as they are backed up.
- Configure output with an error-protection system that allows you to recover data even if there is a read error when you restore.
- Handle AdvFS and UFS sparse files without zero fills.

3.2.2 Backup Level

You can specify the level of incremental backup in the vdump command. A value of 0 specifies complete fileset backup. A higher number specifies a less complete backup. The vdump(8) reference page describes a plan that cycles dump levels so that complete backups are saved on two tapes at a time.

Quota files and fileset quotas are saved only when you are root user and you specify level 0 backup. You can only back up quotas for locally-mounted systems.

The vdump command operates by checking the file modification date. This might cause problems on a subsequent incremental backup because the file modification date does not change if you rename or move a file and do not modify the data. Thus, if you back up your files then move or rename them, change the modification date using the touch command:

```
touch file_name
```

3.2.3 Dumping to Tape

You can place more than one saveset on a single tape by using the vdump command. Set the -N option to specify no rewind or specify a no-rewind device such as /dev/ntape/tape0. This ensures that the tape does not rewind when the vdump command finishes. Issuing another vdump command causes the next saveset to be stored starting at the current tape position.

To restore data from a tape containing more than one saveset, see Section 3.5.4.

If your saveset requires more than one tape to complete, the vdump command prompts you to mount another.

3.2.4 Dumping to Standard Output

When the backup saveset device is the dash (-) character , the vdump command writes to standard output. Thus, you can use the vdump and vrestore commands in a pipeline expression to copy one fileset to another. The following are typical commands; they are equivalent:

```
# vdump -0f - /usr | vrestore -xf - -D /mnt
```

```
# vdump -0 -f - /usr | (cd /mnt; vrestore -x -f -)
```

The rvdump and rvrestore commands are unable to use the dash (-) character. The output device must be specified.

3.2.5 Dumping Subdirectories

You can selectively back up individual subdirectories of a fileset by specifying the subdirectory by using the vdump -D command. Without the -D option, if you specify a subdirectory instead of a fileset on the command line, the vdump command backs up the entire fileset that contains the named subdirectory. If you specify the -D option, backup is always run at level 0.

3.2.6 Dumping to a File or Disk Partition

You can store your saveset as a file on disk. This feature is useful if you want to do full weekly backups to tape for archiving and incremental backups to disk for quick access. Saving to disk increases the speed of saving and recovery because disks are faster than tape devices.

For example, Monday evening you can save an incremental backup to a file containing Monday's activity. In this example, projects.Monday is the saved file in the fileset backups that contains the saveset for the fileset mounted on /projects.

```
# vdump -9f /backups/projects.Monday /projects
```

To restore the copy of a file, revenue, saved to disk in the backup on Monday:

```
# vrestore -xf /backups/projects.Monday -D /projects revenue
```

You can also store your saveset on an empty partition. The saveset will not be available under file system control and could be lost if the partition is reassigned. For example, to dump the fileset mounted at /projects to the partition /dev/disk/dsk2g:

```
# vdump -f /dev/disk/dsk2g -D /projects
```

Do not back up to the a or c partition because these partitions include block 0, which contains the disk label. The device driver does not overwrite the disk label. If you use the a or c partition, an error message is displayed only if you are using a character or raw device; block special devices does not return an error. It is better to back up to another disk partition that does not contain block 0 if you have the space.

If you want to use the a or c partition and the rest of the disk is empty, you must first clear the disk label using the disklabel -z command. All prior information that existed on the disk is lost. If you do not clear the disk label, the vdump command might appear to save valid savesets, but when the vrestore command attempts to interpret the disk label as part of the saveset, it returns the following message:

```
vrestore: unable to use save-set; invalid or corrupt format
```

This is also true for the `rvdump` and `rvrestore` commands.

See `disklabel`(8), `vdump`(8), `vrestore`(8), and *System Administration* for more information.

If you want to copy a complete disk, see *System Administration*.

3.2.7 Compressing Savesets

You can compress savesets as they are backed up. This reduces the amount of storage required for the backup and allows the dump to run faster on slow devices because less data is written. Use the `vdump -C` command to request compression. You cannot specify the compression ratio; it is determined by the contents of the dump.

Note

> If you are using a tape drive that automatically does hardware compression, using the `vdump` command with compression might result in a larger saveset than expected. Sometimes, due to compression algorithms, already compressed data gets expanded when an attempt is made to do more compression.

3.2.8 Dumping with Error Protection

You can use the `vdump -x` command to place **checksum** blocks on your tape so that the `vrestore` command can recover damaged blocks. The `vdump` command creates these blocks every *n* number of blocks you specify. The valid range of *n* is 2 to 32; the default is 8. If the `vrestore` command detects a read error in a block, it uses the other blocks and the checksum block to recreate the bad block.

Dumping with error protection requires saving one extra block for every *n* blocks. It can correct only one block in each series of *n* blocks when the blocks are restored. This means there is a trade-off:

- If you believe tapes are error prone or if you require extremely accurate backups and you have many tapes available for backup, set the value of the `-x` option to 2. This permits error correction of one bad block for every two blocks saved. It requires 50% more tape because after every two dump blocks, a checksum block is written.
- If you believe that tapes are generally reliable but you want to be able to correct a rare bad block, set the value of the `-x` option to 32. This requires 3% more tape because an extra block is added for every 32

blocks written. You can then recover information from any one bad block in the group of 32 dump blocks.

3.2.9 Listing Saved Files

You can check your saveset and make sure you have backed up the files you intended. After your backup is complete, run the `vrestore -t` command to display the files you have saved. This does not initiate the restore procedure.

3.2.10 Dumping and Restoring Files Remotely

The `rvdump` command backs up files from a single mounted fileset or an AdvFS fileset clone to a remote storage device. You must be able to execute the `rsh` command on the remote node to which you are dumping. See `rsh`(8) for server and client access rules.

The `rvdump` command has the same options as the `vdump` command, but you must specify the node name for the device that you are backing your files to. The following example dumps a fileset `sar` to a tape on node `rachem`:

```
# rvdump -0f rachem:/dev/tape/tape0 /sar
```

To restore the fileset from the remote tape drive, enter:

```
# rvrestore -xf rachem:/dev/tape/tape0 -D /sar
```

3.3 Cloning for Online Backup

An AdvFS fileset clone is a read-only snapshot of the data in an existing fileset. If you have the optional AdvFS Utilities, you can create a clone. You must be the root user to clone the root fileset.

A clone does not contain all the data at the time it is created. Rather, it is an outline of the data structure. Data in the original fileset that remains unchanged over the life of the clone is never saved in the fileset clone. When you modify the data in your original files, AdvFS saves the data that existed in the original, page by page, into the clone.

You can create a fileset clone for any AdvFS fileset with the command-line interface (see Section 1.7.11) or with the AdvFS GUI (see Section D.4.3). You cannot clone UFS file systems. Only one clone can exist per fileset.

A clone is useful if the files in your system are changing during the time you want to do your backup. It reflects the state of your system at the moment it was created and backing it up does not interfere with current processing.

Note

After you finish your backup, delete the clone. Clones of active filesets continue to grow as the files are changed.

The following example backs up the pssm fileset on line by creating the pssm_clone fileset and backing it up to the default device. The domain in this example is domain1.

```
# clonefset domain1 pssm pssm_clone
# mkdir /pssm_clone
# mount -t advfs domain1#pssm_clone /pssm_clone
# vdump -0 -u -C /pssm_clone
```

To remove the pssm_clone fileset, enter:

```
# umount /pssm_clone
# rmfset domain1#pssm_clone
```

3.4 Cloning to Back Up Databases

If your database has an online backup utility, use it to back up the database. If it does not, you can back up databases with database down time limited to the short time it takes to create the fileset clone. Backing up a database with an AdvFS fileset clone is the same as backing up any other fileset. You get the same benefits (see Section 3.3).

To back up a database with a clone:

1. Shut down the database so that all database buffers are flushed and the fileset has a complete, consistent copy of the database files.
2. Clone the fileset and mount the clone.
3. Reactivate the database.
4. When you want to back up the fileset clone, run a backup procedure.
5. Unmount and delete the clone.

Caution

Use only the database's own utilities to back up an active database. You can use the vdump and vrestore commands on a fileset clone that contains the database.

If your database files are spread over multiple AdvFS filesets, you should create a clone on each fileset at the same time. This ensures a consistent back up of all the database information.

3.5 Restoring Data

The vrestore command restores files by processing the blocks from a saveset created by using the vdump command. The vrestore command does not work on a saveset created by the UFS dump command.

You do not have to be root user to run the vrestore command, but you must have write privilege for the directory you restore to. Only the root user can restore quota files and fileset quotas. See vrestore(8) for details.

3.5.1 Unique Features of the vrestore Command

The vrestore command performs a number of activities that the UFS restore command does not. You can:

- Display the current vrestore version number.
- Display the source directory path.
- List the saveset structure.
- Display error messages only. Information messages are not be shown.
- Specify how the vrestore command should proceed if it encounters a file that already exists. You can choose whether the command always overwrites an existing file, never overwrites an existing file, or queries you for each event.

3.5.2 Restoring Files

The vrestore command allows you to select specific files and directories to be restored. It can restore data from a file, a pipe, magnetic tapes, or disks.

Use the same version of the vdump and vrestore utilities. If your version of the vrestore utility is unable to read the format of your saveset, you get an error message.

Before you run the vrestore command, you can list the names and sizes of all files in the saveset by running the command using the -t option. The restore operation is not performed.

You can also display the files and directories saved by running the vrestore -i command. You can select individual files or directories to restore.

Restoring data from an AdvFS fileset clone is the same as restoring data from any other fileset.

Start with the full backup if you are restoring an entire fileset. Then restore later incremental backups on top of this to retrieve files that have changed since the full backup was created. Files that were deleted after the full backup was performed are restored. It is necessary to delete these files manually.

3.5.3 Restoring Quotas

You can restore AdvFS user and group quota files either to an AdvFS fileset or to a UFS file system. If you are restoring AdvFS quota files to a UFS file system, quotas must be activated on the UFS file system. AdvFS fileset quotas cannot be restored to a UFS file system because there is no UFS analog to AdvFS fileset quotas. You must be root user to restore quotas.

3.5.4 Restoring from a Tape with Many Savesets

To restore to the current working directory from a tape containing multiple savesets, use the `mt fsf n` command (forward space *n* savesets or files) option to locate the saveset you want to restore. Then use the `vrestore` command.

The following example selects and restores the fourth saveset on a tape:

```
# mt fsf 3
# vrestore -xf /dev/ntape/tape0
```

If you do not know the location of the saveset directory you want to restore, run the `vrestore -i` command. When you have reached the saveset you want, use the interactive shell to specify the files you want to retrieve.

You can use the `vrestore -x` command followed by the file names to selectively restore files from your saveset. You can specify a destination path other than the current directory for the restored files.

The following example restores the file named `data_file` from the `/mnt/fdump` saveset. It is restored to the `/mnt` directory.

```
# vrestore -f /mnt/fdump -D /mnt -x data_file
vrestore: Date of the vdump save-set: Sat Jun 20 15:27:36 2001
```

If your restore operation requires more than one tape, the `vrestore` utility notifies you to mount another.

3.6 AdvFS and NetWorker

NetWorker for Tru64 UNIX provides scheduled, online, automated backup. Use NetWorker with AdvFS as a comprehensive backup solution. NetWorker can automatically back up multiple servers in a heterogeneous environment. It has a graphical interface and several scheduling options.

If users will access filesets during the backup process, use the AdvFS `clonefset` utility to clone all filesets for backup and mount the fileset clones. (You can create a script to accomplish this task.) Then, set up NetWorker to automatically back up the fileset clones on a convenient schedule.

4
Optimizing and Tuning the AdvFS File System

There are a number of ways to configure and tune your AdvFS file system. Some of the tuning functions are available through a graphical user interface (see Appendix D). The *System Configuration and Tuning* manual provides detailed information on tunable parameters for AdvFS.

This chapter covers the following:

- Section 4.1 explains ways to monitor system performance.
- Section 4.2 discusses improving performance by eliminating frag files.
- Section 4.3 suggests ways to improve transaction log performance.
- Section 4.4 details methods of data logging that might improve performance.
- Section 4.5 explains data cache tuning.
- Section 4.6 discusses direct I/O as a method of increasing performance.
- Section 4.7 suggests system attributes that might be changed.
- Section 4.8 details how and when to defragment a file domain.
- Section 4.9 explains how to defragment a single file.
- Section 4.10 discusses balancing the distribution of data.
- Section 4.11 explains how to move filesets to reduce the strain on system resources.
- Section 4.12 explains how to migrate a file to another volume to improve performance.
- Section 4.13 explains AdvFS file system striping.
- Section 4.14 describes how to control the level of a domain panic.

See *System Configuration and Tuning* and Chapter 1 for more detailed information about allocating domains and filesets effectively.

4.1 Monitoring Performance

There are a number of ways to gather performance information:

- The iostat utility reports I/O statistics for terminals, disks, and the CPU. It displays the number of transfers per second (tps) and bytes per second (bps) in kilobytes. From this you can determine where I/O bottlenecks are occurring. That is, if one device shows sustained high throughput, this device is being utilized more than others. Then you can decide the action that might increase throughput: moving files, obtaining faster volumes, striping files, and so on. You can view I/O statistics with the SysMan Monitoring and Tuning - View Input/Output (I/O) Statistics utility or from the command line (see iostat(1)).
- The advfsstat utility displays detailed information about the activity of filesets and domains over time. You can examine, for example, the activity of the buffer cache, volume reads/writes, and the bitfile metadata table (BMT) record. See advfsstat(8) for more information.
- *Collect for Tru64 UNIX* gathers and displays information for subsystems such as memory, disk, tape, network or file systems. Collect runs on all supported releases of Tru64 UNIX. For more information, contact collect_support@compaq.com.

4.2 Improving Performance by Disabling the Frag File

You can control the allocation of space for files that waste more than 5% of their allocated storage. The page size for AdvFS is 8 KB. Files or parts of files that are less than 8 KB are stored in the frag file for the fileset (see Section 1.3.3). Fragging minimizes wasted space in the fileset. If fragging is turned off, I/O is more efficient, but storage requirements increase.

Filesets have frags turned on by default. You can disable them:

- At fileset creation by using the following command format:

  ```
  mkfset -o nofrag domain_name fileset_name
  ```

- For an existing fileset by using the following command format:

  ```
  chfsets -o nofrag domain_name fileset_name
  ```

Note that frags that already exist continue to exist.

The showfsets command displays the fragmentation status of a fileset. For example:

```
# showfsets dom_1 fset_3
    Id           : 3a3a47cb.000b52a5.2.8006
    Files        :        5,  SLim=      0,  HLim=      0
    Blocks (512) :        0,  SLim=      0,  HLim=      0
    Quota Status : user=off group=off
    Object Safety: off
    Fragging     : on
    DMAPI        : off
```

Disabling or enabling frags in a fileset does not affect existing files. If you want to change the frag status of an existing file, you must:

1. Change the frag status for the fileset to the one you want for your file by using the chfsets command.
2. Copy the file to a new location.
3. Delete the original file and rename the new file to the original name.
4. Optionally, change the frag status back to the original for the fileset by using the chfsets command.

For example, to disable fragging in the arizona fileset in the states domain in order to initiate permanent atomic-write data logging for the existing taxes file (Section 4.4):

```
# chfsets -o nofrag states arizona
# cp taxes tmptaxes
# mv -f tmptaxes taxes
```

4.3 Improving Transaction Log File Performance

Each domain has a transaction log file that keeps track of fileset activity for all filesets in the domain. This requires a high volume of read/write activity to the log file. If the log file resides on a congested disk or bus, or if the domain contains many filesets, system performance can degrade. You can shift the balance of I/O activity so that the log file activity does not use up the bandwidth of the device where you have stored your files.

Monitor performance of the volume with the SysMan Monitoring and Tuning - View Input/Output (I/O) Statistics utility or with the iostat utility. If you have AdvFS Utilities, do one of the following if the volume containing the log file appears to be overloaded:

- Divide the domain into several smaller domains. Because each domain has its own transaction log, each log then handles transactions for fewer filesets.
- Move the transaction log file to a faster or less congested volume.
- Isolate the transaction log file on its own volume.

Moving the transaction log file can also be useful when you are using LSM storage and want to increase reliability by placing your transaction log file on a mirrored volume. For example, if the disk containing the transaction log file crashes, the mirrored log can be accessed.

To move the transaction log file to another volume:

1. Use the showfdmn command to determine the location of the log file. The letter L after the volume number indicates the volume on which the log file resides.
2. Use the switchlog command to move the log file to another volume.

For example, to move the transaction log file for the domain region1:

```
# showfdmn region1
     Id              Date Created     LogPgs Version Domain Name
31bf51ba.0001be10 Wed Feb  9 16:24 2000  512       4 region1

Vol  512-Blks    Free % Used Cmode Rblks Wblks Vol Name
  1L  1787904  885168    52%    on   128   128 /dev/disk/dsk0g
  2   1790096 1403872    22%    on   128   128 /dev/disk/dsk0h
     -------------------------
      3578000 2259040    37%

# switchlog region1 2
# showfdmn region1
     Id              Date Created     LogPgs Version Domain Name
31bf51ba.0001be10 Wed Feb  9 16:24 2000  512       4 region1

Vol  512-Blks    Free % Used Cmode Rblks Wblks Vol Name
  1   1787904  885168    52%    on   128   128 /dev/disk/dsk0g
  2L  1790096 1395680    22%    on   128   128 /dev/disk/dsk0h
     -------------------------
      3578000 2250848    37%
```

Isolating the transaction log file allows all log I/O to be separate from other domain reads and writes. As there is no other activity on the log volume, the log file I/O is not slowed down and does not slow down other domain I/O.

To isolate the transaction log file on its own volume:

1. Add a small partition (volume) to the domain for which you are going to isolate the log file.

   Remember that the I/O load of other partition(s) on the device affects the performance of the entire disk including the log file partition.

   If the remaining partitions are allocated to other domains, there might be more than one transaction log file on the same device. This might not be a problem on a solid state disk but might negate the value of isolating the log file on slower devices.
2. Use the switchlog command to move the log file to another volume.
3. Use the showfdmn command to determine the number of free blocks on the volume with the log file.

4. With the information from the showfdmn command, use the dd command to build a dummy file of the right size.
5. Migrate the dummy file to the volume that contains the log file. This fills the volume completely and leaves no space for other files. Because you never access this file, only the transaction log file is active on the volume.

For example, to isolate the transaction log file for the domain sales:

```
# addvol /dev/disk/dsk9a sales
# switchlog sales 2
# showfdmn sales
     Id               Date Created     LogPgs Version Domain Name
312387a9.000b049f Thu Mar 16 14:24 2000  512       4 sales

Vol  512-Blks    Free % Used Cmode Rblks Wblks Vol Name
  1   2050860 1908016     7%    on   128   128 /dev/disk/dsk10c
  2L   131072  122752     6%    on   128   128 /dev/disk/dsk9a
     -------------------------
      2181932 2030768     7%
```

Allocate all the free blocks on the volume containing the log file to a dummy file, /adv1/foo, then move the data to the log file volume:

```
# dd if=/dev/zero of=/adv1/foo count=122752
122752+0 records in
122752+0 records out
# migrate -d 2 /adv1/foo
```

4.4 Improving Data Consistency

The method you choose to write data to a file can affect what is saved if a machine fails. You can synchronize I/O for writing cached metadata and data to disk and you can turn atomic-write data logging on and off.

You can use the fcntl() command to turn synchronous writes and atomic-write data logging on and off. See fcntl(2) and the *Programmer's Guide* for more information. The following sections describe other ways to do this.

4.4.1 Asynchronous I/O

Write requests, by default, are cached; that is, data is written to the **buffer cache**, not immediately to disk. This method, **asynchronous I/O**, generally gives the highest throughput, in part because multiple writes to the same page can be combined into one physical write to disk. This decreases disk traffic and increases the concurrent access of common data by multiple threads and processes. In addition, delaying the write to disk increases the likelihood that a page is combined with contiguous pages into a single, larger physical write, saving seek time and delays caused by rotational latency.

If a crash occurs, the next time a fileset in the domain is mounted, the completed log transactions are replayed to disk and incomplete transactions are backed out so that the original metadata on disk is restored. These log transactions, by default, save only metadata, not the data written to the file. This means that file sizes and locations on disk are consistent but, depending on when the crash occurred, the user data from recent writes might be out of date. This is a trade-off for the increased throughput gained using this method.

4.4.2 Asynchronous Atomic-Write Data Logging I/O

Asynchronous atomic-write data logging I/O is similar to asynchronous I/O except that the data written to the buffer cache is also written to the log file for each write request. This **atomic-write data logging** is done in 8 KB increments. Eventually the data is also written to the file, meaning that the data is written to disk twice: once to the log file and then to the file. The extra write of the data to the log file can degrade throughput compared with using asynchronous I/O.

If a crash occurs, the data is recovered from the log file when the fileset is remounted. As in asynchronous I/O, all completed log transactions are replayed and incomplete transactions are backed out. Unlike asynchronous I/O, however, the user's data has been written to the log, so both the metadata and the data intended for the file can be restored. This guarantees that each 8 KB increment of a write is either completely written to disk or is not written to disk. Because only completed write requests are processed, obsolete, possibly sensitive data located where the system was about to write at the time of the crash can never be accessed. Disk writes in the wrong order, which might cause inconsistencies in the event of a crash, can never occur.

Another way to prevent access to obsolete data is to use the `chfsets -o objectsafety` command (see Section 1.7.10). Choosing object safety prevents an application from reusing old data, but it does not guarantee complete data recovery.

There are two types of atomic-write data logging: persistent and temporary. Persistent logging sets an on-disk flag for a file so atomic-write data logging persists across mounts and unmounts. Temporary data logging sets an in-memory flag activating atomic-write data logging for all files in the fileset for the duration of the mount.

4.4.2.1 Persistent Atomic-Write Data Logging

To turn persistent atomic-write data logging I/O on and off, use the `fcntl()` function or enter the `chfile -L` command:

chfile -L on *file_name*

chfile -L off *file_name*

If a file has a frag, persistent atomic-write data logging cannot be activated. To activate data logging on a file that has a frag, do one of the following:

- Activate temporary atomic-write data logging, which operates on files with frags.
- Choose the `nofrag` option for the fileset when you access it by using the `mkfset` or `chfsets` command.

  If you disable frag files by using the `chfsets -o nofrag` command, files with existing frags do not change. You can, however, copy a file with frags, delete the original file, and rename the copy to the original file name. The file is then free of frag files (see Section 4.2).
- Append enough bytes to the file to bring it up to the next 8 KB boundary. For example, if `fileb` has 6803 bytes, it is stored in one 7 KB frag. To activate data logging, you can add 1389 bytes so the file terminates on an 8 KB boundary:

  ```
  # dd if=/dev/zero of=fileb bs=1 seek=6803 \
      count=1389  conv=notrunc
  ```

  Files that use persistent atomic-write data logging cannot be memory mapped through the `mmap` system call. See Section 5.5 for information on conflicting file usage.

4.4.2.2 Temporary Atomic-Write Data Logging

Use the `mount -o adl` command to set an in-memory flag that activates temporary atomic-write data logging in a fileset for the duration of the mount. Persistent atomic-write data logging commands take precedence over temporary commands while the file is open.

Any application that has the file open can call the `fcntl()` function to turn off temporary atomic-write data logging, or use the `chfile -L off` command to turn off persistent atomic-write data logging while the file is open. All applications that have the file open are affected. After all applications close the file, temporary atomic-write data logging is restored for the file. You can check atomic-write data logging status for a file by using the `chfile` command.

Files using temporary atomic-write data logging can be memory mapped. Temporary atomic-write data logging is suspended until the last thread using the memory-mapped file unmaps it. In addition, files that have frags can use temporary data logging feature. NFS mounting does not affect logging behavior.

4.4.3 Synchronous I/O

Synchronous I/O is similar to asynchronous I/O, but the data is written both to the cache and to the disk before the write request returns to the calling application. This means that if a write is successful, the data is guaranteed to be on disk. Synchronous I/O reduces throughput because the write does not return until after the I/O is complete. In addition, since the application, not the file system, determines when the data needs to be flushed to disk, the likelihood of consolidating I/Os might be reduced if synchronous write requests are small.

To turn synchronous I/O off and on, use the `O_SYNC` or `O_DSYNC` flag to the `open()` system call (see the *Programmer's Guide*). To force all applications to use synchronous I/O even if files are not opened in that mode, enter the `chfile -l` command:

```
chfile -l on file_name

chfile -l off file_name
```

4.4.4 Synchronous Atomic-Write Data Logging I/O

You cannot use both the `-l` and `-L` options of the `chfile` command to set synchronous atomic-write data logging. However, if you activate persistent atomic-write data logging on a file by using the `chfile -L on` command, you can open the file for synchronous I/O by using the `O_SYNC` or `O_DSYNC` flag to the `open()` system call (see the *Programmer's Guide*).

You can activate temporary atomic-write data logging for the fileset by using the `mount -o adl, sync` command.

4.5 Data Cache Tuning

Caching improves performance when data is reused frequently. AdvFS uses a dynamic memory cache called the **Unified Buffer Cache (UBC)** to manage file metadata and user data.

By using the UBC for caching, AdvFS can maintain file data in memory as long as memory is available. If other system resources require some of the memory in use by the file system cache, the UBC can reclaim the memory that is needed.

Because AdvFS uses the UBC to control caching, the cache is tuned with the UBC tunable parameters. These include:

- Variables that modify the maximum percentage of physical memory that the UBC can use at one time
- The percentage of pages that must be dirty before the UBC starts writing them to disk
- The maximum amount of memory allocated to the UBC that can be used to cache a single file

See *System Configuration and Tuning* for guidelines for modifying these parameters.

Although caching data is the default and generally improves file system performance, in some situations an application can increase throughput by bypassing the data cache (see Section 4.6).

4.6 Improving Data Transfer Rate with Direct I/O

Direct I/O mode bypasses caching and synchronously reads and writes data from a file without copying the data into a cache (the normal AdvFS process). That is, when direct I/O is enabled for a file, read and write requests on it are executed to and from disk storage through direct memory access (similar to raw I/O), bypassing AdvFS caching. This can improve the speed of the I/O process for applications that access data only once.

Although direct I/O handles requests of any byte size, you get the best performance when the requested transfer size is aligned on a disk sector boundary and the transfer size is an even multiple of the underlying sector size (currently 512 bytes).

Direct I/O is particularly suited for files that are used exclusively by a database. However, if an application tends to access data multiple times, direct I/O can adversely impact performance because caching does not occur. When you specify direct I/O, it takes precedence and any data already in the buffer cache for that file is automatically flushed to disk.

To open a file for direct I/O, use the `open()` function and specify the `O_DIRECTIO` flag. For example, for `file_x` enter:

```
open (file_x, O_DIRECTIO|O_RDWR, 0644)
```

Regardless of the previous mode, the new mode is direct I/O and remains so until the last close of the file. Note that direct I/O, atomic-write data logging, and memory mapping are mutually exclusive modes. Therefore, if the file is already open for atomic-write data logging or is memory mapped, then calling the `open` function to initiate direct I/O fails.

The `fcntl()` function can be used to determine whether the file is open in cached or in direct I/O mode. See `fcntl`(2) and `open`(2), or the *Programmer's Guide* for more information.

4.7 Changing Attributes to Improve System Performance

You can change a number of attributes to improve system performance. *System Configuration and Tuning* details the significance of each attribute and the trade-offs engendered when they are changed. See sysconfig(8) for more information. You can modify attributes to:

- Increase the dirty-data caching threshold

  Dirty or modified data is data that was written by an application and cached but is not yet been written to disk. You can modify the amount of **dirty data** that AdvFS caches for each volume in a domain by using the chvol -t command or, for all new volumes of a file system, by using the AdvfsReadyQLim attribute (see chvol(8)).

  Modifying this variable by using the chvol command is most effective if smooth sync is disabled. If your system is using smooth sync (the default), then the rate at which the dirty data is flushed to disk is best tuned by using the smoothsync_age attribute.

- Promote continuous I/O by using the smoothsync_age attribute

  The smoothsync_age attribute specifies the number of seconds that a modified page stays in the buffer cache before being flushed to disk. This allows the file system to balance the need to flush modified pages to disk in a timely manner with the benefits of keeping the page in memory while all modifications are being made.

- Change the I/O transfer size

  AdvFS reads and writes data by 8 KB pages. The maximum transfer size is determined by the device driver and depends on the underlying storage configuration but is typically 128 or 256 blocks. LSM might assign a larger maximum transfer size. You can adjust the maximum transfer size by using the chvol command with the -r (read) or -w (write) options (see chvol(8)).

- Flush modified memory-mapped pages

  The AdvfsSyncMmapPages attribute controls whether modified memory-mapped pages are flushed to disk during a sync system call.

- Increase the memory available for access structures

  AdvFS allocates file access structures until the percentage of pageable memory used for the access structures reaches AdvfsAccessMaxPercent. Increasing the value of the AdvfsAccessMaxPercent attribute might improve AdvFS performance on systems that open and reuse many files, but this decreases the memory available for the virtual memory subsystem and the Unified Buffer Cache (UBC). Decreasing the value of the attribute frees pageable memory but might degrade AdvFS performance on systems that open and reuse many files.

4.8 Defragmenting a Domain

The AdvFS file system attempts to store file data in contiguous blocks on a disk. This collection of contiguous blocks is called a file extent. If all data in a file is stored in contiguous blocks, that file has one file extent. However, as files grow, contiguous blocks on the disk might not be available to accommodate the new data, and the system spreads the file over discontiguous blocks. As a result, the file is fragmented on the disk and consists of multiple file extents. File fragmentation degrades the read/write performance because many disk addresses must be examined to access a file.

The `defragment` utility reduces the amount of file fragmentation in a domain by attempting to make the files more contiguous so that the number of file extents is reduced. In addition, defragmenting a domain often makes the free space on a disk more contiguous so files that are created later are also less fragmented.

Files might be moved to other volumes in the defragmentation of a multivolume domain. You cannot control the placement of files during defragmentation, but you can use the `showfile` to identify where a file is stored. If you want to move a file, use the `migrate` command (see Section 4.12).

You can improve the efficiency of the defragment process by deleting any unneeded files in the domain before running the `defragment` utility. Aborting the defragment process does not damage the file system. Files that have been defragmented remain in their new locations.

It is difficult to specify the load that defragmenting places on a system. The time it takes to defragment a domain depends on:

- The size of the volume(s)
- The amount of free space available
- The activity of the system
- The configuration of your domain

  Because the `defragment` utility creates a thread per volume (up to a maximum of 20 threads), a domain consisting of several small volumes is faster to defragment than one consisting of a large volume. Multiple threads might exact a severe performance penalty for ongoing I/O. If you want to limit defragmentation to a single thread (similar to Version 4 operating system software behavior), use the `defragment -N 1` command.

To defragment a domain, use the SysMan Manage an AdvFS Domain utility, a graphical user interface (see Appendix D), or enter the `defragment` command from the command line:

```
defragment domain_name
```

The following restrictions apply to running the `defragment` command:

- You must have root user privileges.
- All filesets in the domain must be mounted. If you try to defragment an active domain that includes unmounted filesets, an error message is displayed.
- A minimum free space of 1% of the total space or 5 MB per volume (whichever is less) must be available to defragment each volume.
- The `defragment` utility cannot be run while the `addvol`, `rmvol`, `balance`, or `rmfset` command is running in the same domain.

See `defragment`(8) for more information.

4.8.1 Choosing to Defragment

Run the `defragment` utility on your domain when you experience performance degradation and then only when file system activity is low.

To determine the amount of fragmentation in your domain without starting the utility, run the `defragment -v -n` command. If the average number of extents or the number of extents per file with extents is high or the aggregate I/O performance is low, defragmentation might be helpful.

The level of fragmentation you should allow in your file system before running the utility depends on the size of the files and the number of extents. This is largely application dependent, so monitor the number of extents to see if elevated extent counts correlate with decreased application performance. In many cases, even a large, fairly fragmented file does not show a noticeable decrease in performance because of fragmentation. It is not necessary to run the `defragment` command on a system that is not experiencing performance-related problems because of excessive file fragmentation.

If your file system has been untouched for a month or two, that is, if you did not run full periodic backups nor regularly reference your whole file system, it is a good idea to run the `verify` command before you run the `defragment` command. Run the `verify` command when there is low file system activity.

Running the `balance` utility before you run `defragment` might speed up the defragmentation process.

If you have a system, such as a mail server, that contains files that are mostly smaller than 8 KB, run the `defragment` command only when the output of the showfile -x /mntpt/.tags/1 command indicates that the frag file is highly fragmented. Here /mntpt is the mount point for the fileset and .tags/1 is the frag file.

If you find that one file shows high fragmentation, you can defragment that file (see Section 4.9).

4.8.2 Defragmenting Example

If you have the hardware resources and AdvFS Utilities, you can add a volume by using the addvol command then remove the previous volume by using the rmvol command. Removing the volume migrates the domain to the new volume, and the files in it are defragmented as part of the migration.

The following example looks at the fragmentation of the accounts_domain domain and at the number of extents in the orig_file_1 file, and then defragments the domain for a maximum of 15 minutes. Verbose mode is used to display the fragmentation data at the beginning of each pass through the domain and at the end of the defragmentation process.

```
# defragment -vn accounts_domain
defragment: Gathering data for ’accounts_domain’
Current domain data:
   Extents:                 263675
   Files w/ extents:        152693
   Avg exts per file w/exts:  1.73
   Aggregate I/O perf:         70%
   Free space fragments:     85574
                <100K   <1M   <10M   >10M
   Free space:   34%   45%    19%     2%
   Fragments:  76197  8930    440      7
# showfile -x orig_file_1
    Id Vol PgSz Pages XtntType Segs SegSz   I/O Perf  File
6.8002   2   16    71   simple   **    ** async  82%  orig_file_1
        \\: 1
            pageOff    pageCnt    vol    volBlock    blockCnt
                  0          5      2       40720          80
                  5         12      2       41856         192
                 17         16      2       40992         256
                 33          7      2       42048         112
                 40         12      2       41360         192
                 52         15      2       42160         240
                 67          4      2       41792          64
            extentCnt: 7
# defragment -v -t 15 accounts_domain
defragment:  Defragmenting domain ’accounts_domain’

Pass 1;
  Volume 2: area at block      144 (  130800 blocks): 0% full
  Volume 1: area at block   468064 (  539008 blocks): 49% full
  Domain data as of the start of this pass:
    Extents:                   7717
    Files w/extents:           6436
    Avg exts per file w/exts:  1.20
    Aggregate I/O perf:         78%
    Free space fragments:       904
                    <100K    <1M    <10M    >10M
     Free space:      4%      5%    12%     79%
     Fragments:      825     60      13       6
Pass 2;
  Volume 1: area at block   924288 (  547504 blocks): 69% full
  Volume 2: area at block      144 (  130800 blocks):  0% full
  Domain data as of the start of this pass:
    Extents:                   6507
    Files w/extents:           6436
    Avg exts per file w/exts:  1.01
    Aggregate I/O perf:         86%
    Free space fragments:      1752
                    <100K    <1M    <10M    >10M
     Free space:      8%     13%    11%      67%
     Fragments:     1574     157      15       6

Pass 3;
  Domain data as of the start of this pass:
    Extents:                   6485
    Files w/extents:           6436
    Avg exts per file w/exts:  1.01
    Aggregate I/O perf:         99%
    Free space fragments:       710
                    <100K    <1M    <10M    >10M
     Free space:      3%   11%      21%     65%
     Fragments:      546   126       32       6

Defragment: Defragmented domain 'accounts_domain'
```

Information displayed before each pass and at the conclusion of the defragmentation process indicates the amount of improvement made to the domain. A decrease in the `Extents` and `Avg exts per file w/extents` values indicates a reduction in file fragmentation. An increase in the `Aggregate I/O perf` value indicates improvement in the overall efficiency of file-extent allocation.

4.9 Defragmenting a File

To determine the amount of fragmentation level of a single file, run the `showfile -x` command to show the extents in the file. (You might already have this information from examining the output of the `defragment -v -n` command.)

To reduce the fragmentation of a file:

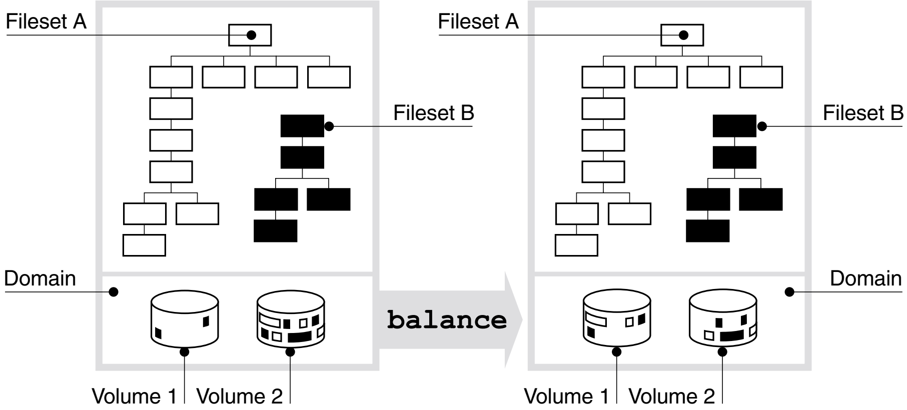
- Use the `migrate` utility to move the file to the same or a different volume containing adequate contiguous free space.
- Back up and restore a file. This tends to defragment it.

    1. Back up the file by using the `vdump` command.
    2. Delete or rename the file.
    3. Restore the data by using the `vrestore` command.

4.10 Balancing a Multivolume Domain

The `balance` utility distributes the percentage of used space evenly between volumes in a multivolume domain created with the optional AdvFS Utilities. This improves performance and evens the distribution of future file allocations.

Figure 4–1: Balancing a Domain



Files are moved from one volume to another, as illustrated in Figure 4–1, until the percentage of used space on each volume in the domain is as equal as possible. Because the `balance` utility does not generally split files, domains with very large files might not balance as evenly as domains with smaller files.

To redistribute files across volumes, use the SysMan Manage an AdvFS Domain utility, a graphical user interface (see Appendix D), or enter the `balance` command from the command line:

```
balance domain_name
```

If you interrupt the balance process, all relocated files remain at their new locations. The rest of the files remain in their original locations.

The following restrictions apply to running the balance utility:

- You must have root user privileges.
- All filesets in the domain must be mounted. If you try to balance an active domain that includes unmounted filesets, an error message is displayed.
- The balance utility cannot run while the addvol, rmvol, defragment, or rmfset command is running in the same domain.

See balance(8) for more information.

4.10.1 Choosing to Balance

Use the showfdmn command to display domain information. Look at the % Used field to determine if the files are evenly distributed.

Use the balance utility to even file distribution after you have added a volume by using the addvol command or removed a volume by using the rmvol command (if there are multiple volumes remaining).

4.10.2 Balance Example

In the following example, the multivolume domain usr_domain is not balanced. Volume 1 has 63% used space while volume 2, a smaller volume, has 0% used space (it has just been added). After balancing, both volumes have approximately the same percentage of used space.

```
# showfdmn usr_domain
            Id       Date Created      LogPgs Version Domain Name
3437d34d.000ca710 Mon Apr 3 10:50:05 2000 512       4 usr_domain

 Vol   512-Blks   Free % Used  Cmode Rblks  Wblks  Vol Name
  1L   1488716 549232    63%     on   128    128  /dev/disk/dsk0g
  2     262144 262000     0%     on   128    128  /dev/disk/dsk4a
    --------- -------  ------
      1750860 811232    54%
# balance usr_domain
 balance: Balancing domain ’usr_domain’
 balance: Balanced domain ’usr_domain’
# showfdmn usr_domain
            Id       Date Created      LogPgs Version Domain Name
3437d34d.000ca710 Mon Apr 3 10:50:05 2000 512       4 usr_domain

 Vol   512-Blks   Free % Used  Cmode Rblks  Wblks  Vol Name
  1L   1488716 689152    54%     on   128    128  /dev/disk/dsk0g
  2     262144 122064    53%     on   128    128  /dev/disk/dsk4a
    --------- -------  ------
      1750860 811216    54%
```

4.11 Moving Filesets to Different Volumes

If you suspect that a fileset or domain is straining system resources, run the iostat utility either from the SysMan Monitoring and Tuning - View Input/Output (I/O) Statistics utility, or from the command line (see iostat(1)). If the filesets or domains are located on devices that appear to be a bottleneck, you can **migrate** files or pages of files to equalize the load. If a high-performance device is available, you can move a file that is I/O-intensive to the more efficient volume.

To move a domain and its fileset to a new volume:

1. Make a new domain on the new device. It must have a temporary new name.
2. Create a fileset with the same name as the old.
3. Create a temporary mount-point directory for the fileset.
4. Mount the new fileset on the temporary mount point.
5. Use the vdump command to copy the fileset from the old device. Use the vrestore command to restore it to the newly mounted fileset.
6. Unmount the old and new filesets.
7. Rename the new domain to the old name. Since you have not changed the domain and fileset names, it is not necessary to edit the /etc/fstab file.
8. Mount the new fileset using the mount point of the old fileset. The directory tree is then unchanged. Delete the temporary mount-point directory.

If you have more than one fileset in your domain, follow steps two through eight for each fileset.

If you are running operating system software Version 5.0 or later, the new domain is created with the new DVN of 4 (see Section 1.6.3). However, if you must retain the DVN of 3 in order to use earlier versions of the operating system, see mkfdmn(8). The vdump and vrestore utilities are not affected by the change of DVN.

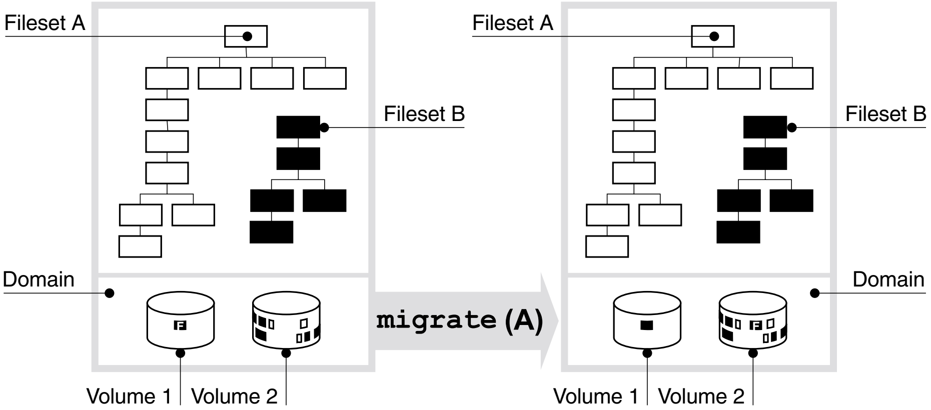
The following example moves the domain accounts with the fileset technical to volume dsk3c. The domain new_accounts is the temporary domain and is mounted initially at /tmp-mnt. Assume the fileset accounts#technical is mounted on /technical. Assume that the /etc/fstab file has an entry instructing the system to mount accounts#technical on /technical.

```
# mkfdmn /dev/disk/dsk3c new_accounts
# mkfset new_accounts technical
# mkdir /tmp_mnt
# mount new_accounts#technical /tmp_mnt
# vdump -dxf - /technical|vrestore -xf - -D /tmp_mnt
# umount /technical
# umount /tmp_mnt
# rmfdmn accounts
# rmdir /tmp_mnt
# mv /etc/fdmns/new_accounts/ /etc/fdmns/accounts/
# mount accounts#technical /technical
```

4.12 Migrating Files to Different Volumes

If you have the optional AdvFS Utilities, you can use the `migrate` utility to move heavily accessed or large files to a different volume in the domain. The `balance` and `defragment` utilities also migrate files but are not under user control. You can specify the volume where a file is to be moved or allow the system to pick the best space in the domain by using the `migrate` command. You can migrate either an entire file or specific pages to the same or a different volume. Figure 4–2 illustrates the migration process.

Figure 4–2: Migrating Files



To move an entire file to a specific volume, use the `migrate -d` command:

migrate -d *destination_vol_index file_name*

A file that is migrated is defragmented in the process if possible. This means that you can use the `migrate` command to defragment selected files.

The following restrictions apply to the `migrate` utility:

- You must have root user privilege.
- You can perform only one migrate operation at a time on the same file.

- When you migrate a striped file, you can migrate from only one volume at a time.
- The migrate utility does not evaluate your migration decisions. For example, you can move more than one striped file segment to the same disk, which defeats the purpose of striping the file.

4.12.1 Choosing to Migrate

Choose the migrate utility over the balance utility when you want to control the movement of individual files. The balance utility moves files only to optimize distribution. For example, it might move many small files when moving a single larger file would be a better solution for your system.

Choose the migrate utility over the defragment utility when you want to defragment an individual file. If you have a large enough contiguous area on disk, you can migrate the file to that area to defragment it.

You can use the showfile -x command to look at the extent map and the performance of a file. A low performance rate (less than 80%) indicates that the file is fragmented on the disk. The extent map shows whether the entire file or a portion of the file is fragmented.

4.12.2 Migrate Example

The following example displays the extent map of a file called src and migrates the file. The file, which resides in a two-volume domain, shows a change from 11 file extents to one and a performance efficiency improvement from 18% to 100%.

```
# showfile -x src
    Id Vol PgSz Pages XtntType  Segs  SegSz  I/O  Perf  File
8.8002   1   16    11   simple    **     ** async  18%   src
             extentMap: 1
        pageOff    pageCnt     vol    volBlock    blockCnt
              0          1       1      187296          16
              1          1       1      187328          16
              2          1       1      187264          16
              3          1       1      187184          16
              4          1       1      187216          16
              5          1       1      187312          16
              6          1       1      187280          16
              7          1       1      187248          16
              8          1       1      187344          16
              9          1       1      187200          16
             10          1       1      187232          16
        extentCnt: 11

# migrate -d 2 src
# showfile -x src
```

```
    Id Vol PgSz Pages XtntType Segs SegSz  I/O  Perf  File
8.8002   1   16    11   simple   **    ** async 100%  src
   extentMap: 1
      pageOff    pageCnt     vol    volBlock    blockCnt
            0         11       2       45536         176
      extentCnt: 1
```

The src file now resides on volume 2 of the domain, consists of one file extent, and has a 100% performance efficiency. Note that in the command output, the first data line of the display lists the metadata. The metadata does not migrate to the new volume. It remains in the original location. The extentMap portion of the display lists the file's migrated pages.

4.13 Striping Files

You can **stripe**, that is, distribute, files across a number of volumes. This increases the sequential read/write performance because I/O requests to the different disk drives can be overlapped. Virtual storage solutions, such as LSM, RAID, and storage area networks (SAN), stripe all files and are usually configured at system setup. AdvFS striping is applied to single files and can be executed at any time.

Note

Use AdvFS striping only on directly attached storage that does not include LSM, RAID, or a SAN. Combining AdvFS striping with system striping might interfere with optimal placement and cause system degradation.

The AdvFS stripe utility distributes **stripe segments** across specific disks (or volumes) of a domain. You must have the AdvFS Utilities to run this command. The stripe width is fixed at 64 KB, but you can specify the number of volumes over which to stripe the file.

The form of the stripe command is:

stripe -n *volume_count filename*

To stripe a file:

1. Create a new, empty file.

   If you do not create an empty file, the following error message is displayed:

   ```
   stripe: advfs_set_bf_attributes failed-ENOT_SUPPORTED (-1041)
   ```

2. Stripe it across the number of volumes desired.
3. Copy the data from the original file to the striped file.

4. Delete the original file and rename the striped file, if desired.

As the file is appended, AdvFS determines the number of pages per stripe segment; the segments alternate among the disks in a sequential pattern. For example, the file system allocates the first segment of a two-disk striped file on the first disk and the next segment on the second disk. This completes one sequence, or stripe. The next stripe starts on the first disk, and so on. Because AdvFS spreads the I/O of the striped file across the specified disks, the sequential read/write performance of the file increases.

You cannot use the `stripe` utility to modify the number of disks that an already striped file crosses or to restripe a file that is already striped. To change the configuration of a striped file, you must create a new file, stripe it, then copy the original file data to it.

You cannot use AdvFS stripe utility on the `/etc/fstab` file.

4.13.1 Choosing to Use AdvFS Striping on a File

Before you use the `stripe` utility, run the `iostat` utility either from the SysMan Monitoring and Tuning - View Input/Output (I/O) Statistics utility, or from the command line to determine if disk I/O is causing the bottleneck. See `iostat`(1) for more information. Cross check the blocks per second and transactions per second with the drive's sustained transfer rate. Maximum stripe performance can be achieved if each stripe disk is on its own disk controller.

Using AdvFS striping when system-wide striping is in effect might degrade performance.

4.13.2 AdvFS Stripe Example

The following example creates an empty file, `file_1`, stripes it, copies data from the original file, `orig_file_1`, into the striped file, then displays the extents of the striped file:

```
# touch file_1
# ls -l file_1
-rw-r--r-- 1 root  system 0 Oct 07 11:06 file_1
# stripe -n 3 file_1
# cp orig_file_1 file_1
# showfile -x file_1

     Id Vol PgSz Pages XtntType Segs SegSz I/O   Perf File
7.8001   1   16    71   stripe    3     8 async 100% file_1
  extentMap: 1
     pageOff   pageCnt   volIndex  volBlock   blockCnt
           0         8          2     42400        384
          24         8
```

```
        48         8
   extentCnt: 1
extentMap: 2
   pageOff   pageCnt   volIndex   volBlock   blockCnt
         8         8          3      10896        384
        32         8
        56         8
   extentCnt: 1

extentMap: 3
   pageOff   pageCnt   volIndex   volBlock  blockCnt
        16         8          1     186784       368
        40         8
        64         7
   extentCnt: 1
```

4.13.3 Removing AdvFS Striping

You can alter the pattern of striping in your domain by:

- Removing striping from a file

  Copy the striped file to a file that is not striped. Delete the original.

- Removing a striped volume

  If you remove a volume that contains an AdvFS stripe segment, the `rmvol` utility moves the segment to another volume that does not already contain a stripe segment of the same file. If all remaining volumes contain stripe segments, the system requests confirmation before the segment is moved to a volume that already contains a stripe segment of the file. To retain the full benefit of striping when this occurs, stripe a new file across existing volumes and copy the file with the doubled-up segments to the new file.

4.14 Controlling Domain Panic Information

The `AdvfsDomainPanicLevel` attribute allows you to choose whether to have crash dumps created when a domain panic occurs. Values of the attribute are:

- 0 – Create crash dumps for no domains.
- 1 – Create crash dumps only for domains with mounted filesets (default).
- 2 – Create crash dumps for all domains.
- 3 – Promote the domain panic to a system panic. The system crashes.

See `sysconfig`(8) for information on changing attributes. See Section 5.8.6 for information about recovering from a domain panic.

5

Troubleshooting

This chapter examines problems that, while universal for file systems, might have solutions unique to AdvFS. See *System Configuration and Tuning* for related information about diagnosing performance problems.

This chapter covers the following:

- Section 5.1 details the commands that you can use to check disk usage.
- Section 5.2 suggests preventative maintenance strategies.
- Section 5.3 explains how to increase the size of the root domain.
- Section 5.4 identifies operating system incompatibilities.
- Section 5.5 explains I/O method incompatibilities.
- Section 5.7 lists ways that you can improve poor performance.
- Section 5.8 suggests ways to fix disk problems.
- Section 5.9 explains how to restore the `/etc/fdmns` directory.
- Section 5.10 examines how to recover from volume failure.
- Section 5.11 explains how to recover from the failure of the root domain.
- Section 5.12 details how to restore a multivolume domain.
- Section 5.13 suggests methods of crash recovery.

5.1 Checking Free Space and Disk Usage

You can look at the way space is allocated on a disk by file, fileset, or domain. Table 5–1 describes command-line commands that you can use to examine disk space usage.

Table 5–1: Disk Space Usage Commands

Command	Description
`df`	Displays disk space usage by fileset. Available space for a fileset is limited by the fileset quota if it is set.
`du`	Displays information about block allocation for files. Use the `-a` option to display information for individual files.

Table 5–1: Disk Space Usage Commands (cont.)

Command	Description
`ls`	Displays the space used by files. The `-l` option shows the space spanned by a sparse file. The `-s` option shows the actual block usage and might be more useful for use with sparse files.
`showfdmn`	Displays the attributes and block usage for each volume in an active domain. For multivolume domains, additional volume information is displayed.
`showfile`	Displays block usage and volume information for a file or for the contents of a directory.
`showfsets`	Displays information about the filesets in a domain. Use to display fileset quota limits.
`vdf`	Displays used and available disk space for a fileset or a domain.

See the reference pages for the commands for more complete information.

Under certain conditions, the disk usage information for AdvFS can become corrupt. Run the `quotacheck -v` command to correct the disk usage information.

5.2 Preventative Maintenance

This section describes a number of things you can do to prevent problems with your AdvFS file system.

5.2.1 Failing Disks

Back up your data regularly and frequently, and watch for signs of impending disk failure. Try to remove files from a problem disk before it fails. See the Event Management information in *System Administration* for more information about examining disk activity.

5.2.2 Verifying File System Consistency

To ensure that metadata is consistent, run the `verify` command to verify the file system structure. The `verify` utility checks disk structures such as the bitfile metadata table (BMT), the storage bitmaps, the tag directory, and the `frag` file for each fileset. It verifies that the directory structure is correct, that all directory entries reference a valid file, and that all files have a directory entry. You must be the root user to run this command.

It is a good idea to run the `verify` command:

- When problems are evident (corruptions, domain panic, lost data, I/O errors)
- Before an update installation
- If your files have not been accessed in three to six months or longer, and you plan to run utilities such as `balance`, `defragment`, `migrate`, `quotacheck`, `repquota`, `rmfset`, `rmvol`, or `vdump` that access every file in a domain

Use the SysMan Manage an AdvFS Domain utility, or enter the `verify` command from the command line:

`verify` *`domain_name`*

The `verify` command mounts filesets in special directories. If the `verify` command is unable to mount a fileset due to the failure of a domain, as a last resort you can run the `verify -F` command. The `-F` option mounts the fileset using the `-d` option of the `mount` command, which means that AdvFS initializes the transaction log file for the domain without recovering the domain.

Caution

> Because no domain recovery occurs for previously incomplete operations, using the `verify -F` command could cause data corruption.

Under some circumstances the `verify` command might fail to unmount the filesets. If this occurs, you must unmount the affected filesets manually and delete the mount points that were created in the `/etc/fdmns/<domain_name>` file.

On machines with millions of files, sufficient swap space must be allocated for the `verify` utility to run to completion. If the amount of memory required by the `verify` utility exceeds the kernel variable `proc/max_per_proc_data_size` process variable, the utility does not complete. To overcome this problem, allocate up to 10% of the domain size in swap space for running the `verify` command.

The following example verifies the `domainx` domain, which contains the filesets `setx` and `sety`:

```
# verify domainx
+++Domain verification+++
Domain Id 2f03b70a.000f1db0
Checking disks ...
Checking storage allocated on disk /dev/disk/dsk10g
Checking storage allocated on disk /dev/disk/dsk10a
Checking mcell list ...
Checking mcell position field ...
Checking tag directories ...

+++ Fileset verification +++
+++ Fileset setx +++
Checking frag file headers ...
Checking frag file type lists ...
Scanning directories and files ...
     1100
Scanning tags ...
     1100
Searching for lost files ...
     1100

+++ Fileset sety +++
Checking frag file headers ...
Checking frag file type lists ...
Scanning directories and files ...
     5100
Scanning tags ...
     5100
Searching for lost files ...
     5100
```

In this example, the `verify` utility finds no problems with the domain. See `verify`(8) for more information.

5.3 Increasing the Size of an AdvFS root Domain

The AdvFS root domain is limited to one volume (partition) unless you are running a cluster configuration. If you want to increase the size of the root domain, you must recreate the root domain on a larger volume. This section explains how to recreate the root domain on a different device. It does not cover the case of repartitioning the current root volume and restoring root to it. If you are moving the root domain to another disk already installed in the system, you can skip the section on installing the disk and begin at Section 5.3.2.

You need the following to move your root domain:

- Current operating system CD-ROM

  You can use the operating system CD-ROM that is packaged with the distribution media to recreate your root domain.

  If your local site provides a Remote Installation Service (RIS) server, you can boot your system across the network. If you choose RIS services, follow your site-specific procedures and consult the *Installation Guide*.

- Backup device

You will need either a backup tape or an unused disk partition to backup the root domain.

- Information about console commands

  You will use Alpha System Reference Manual (SRM) console commands at the system console prompt (`>>>`) to perform some tasks. These commands are documented in the hardware manual for your Alpha system. If you cannot find the printed document, it is usually shipped as a printable file on a CD-ROM supplied with the system.

This section explains increasing the size of a root domain on a non-clustered system. For other configurations, see *System Administration* and *Cluster Administration*. If your root volume is an LSM volume, see *Logical Storage Manager*.

5.3.1 Installing a New Disk for the root Domain

To move your root domain to a new disk, you must first install the disk and have it recognized.

1. Log in as root and shut down the system.

   ```
   # shutdown -h now
   ```

2. Add the new disk device. For more information see your hardware manuals.

3. Verify that the SRM console recognizes the newly added disk. In this example, `DKB300` (an RZ1BB-CS) was added.

   ```
   >>> show device
   polling ncr0(NCR 53C810) slot 1, bus 0 PCI,
        hose 1 A SCSI Bus ID 7
   dka500.5.0.1.1     DKA500                    RRD45     1645
   polling isp0(QLogic ISP1020) slot 4, bus 0 PCI,
        hose 1 SCSI Bus ID 7
   dkb0.0.0.4.1       DKB0                      RZ1DB-CA  LYJ0
   dkb100.1.0.4.1     DKB100                    RZ1CB-CA  LYJ0
   dkb200.2.0.4.1     DKB200                    RZ1CB-CA  LYJ0
   dkb300.3.0.4.1     DKB300                    RZ1BB-CS  0656
   mkb400.4.0.4.1     MKB400                    TLZ10     02ab
   ...
   ```

4. Boot the original system disk to update the device information databases with the new device. In this example, the default boot device, `dkb0`, is booted.

   ```
   >>> show bootdef_dev
   bootdef_dev            dkb0.0.0.4.1
   >>> boot
   ```

During the boot process, the operating system recognizes the new device and updates the device information databases accordingly.

```
...
dsfmgr: NOTE: updating kernel basenames for system at /
 scp kevm tty00 tty01 lp0 random urandom dmapi dsk3 dsk4
 dsk5 floppy0 cdrom0 -dsk6a +dsk6a -dsk6a +dsk6a -dsk6b
+dsk6b -dsk6b +dsk6b -dsk6c +dsk6c -dsk6c +dsk6c -dsk6d
+dsk6d -dsk6d +dsk6d -dsk6e +dsk6e -dsk6e +dsk6e -dsk6f
+dsk6f -dsk6f +dsk6f -dsk6g +dsk6g -dsk6g +dsk6g -dsk6h
+dsk6h -dsk6h +dsk6h
...
```

In this example, the operating system's device name for the added disk is `dsk6`.

5.3.2 Configuring a Device for Use as the root Volume

To make the device available to be used as the root volume, you must configure, that is, label and partition, it. You must be the root user to perform this operation. For methods of labeling your disk, see *System Administration*, "Partitioning Disks Using diskconfig" and "Manually Partitioning Disks."

Be sure to specify AdvFS for the Boot Block. Use the `disklabel` command with the `-t advfs` option or, if you are using the `disconfig` utility, choose AdvFS from the `Boot Block:` list.

Caution

Modifying a disk's partition layout destroys some or all of the data on disk. Be certain that you do not need any data on the disk that you choose for the new root domain.

For example, if you have expanded the `a` partition to 500 MB (1024000 512–byte sectors) and allocated the remaining space on the disk to the `b` partition as swap, your disk label might look like the following:

```
# disklabel dsk6
 #      size  offset fstype fsize bsize cpg # ~Cyl values
a:   1024000       0 unused     0     0     #   0 - 744*
b:   3086480 1024000 swap       0     0     # 744*- 2987*
c:   4110480       0 unused     0     0     #   0 - 2987*
d:         0       0 unused     0     0     #   0 - 0
e:         0       0 unused     0     0     #   0 - 0
f:         0       0 unused     0     0     #   0 - 0
g:   1858632  393216 unused     0     0     # 285*- 1636*
h:   1858632 2251848 unused     0     0     # 636*- 2987*
```

5.3.3 Backing up the Current root Domain

The first step in moving a root domain is to make a full backup of the domain. Use a backup tape or an unused disk partition.

For example, to back up the root domain to tape /dev/tape/tape0_d1:

```
# vdump -0 -f /dev/tape/tape0_d1 /
```

To back up the root domain to an unused partition, create a temporary domain, fileset and mount-point directory. Back up to a file in that fileset. For example, for the domain TMP_BACKUP, the fileset tmp_backup, the mount point /tmp_backup, and the file containing the dump, root_backup.vdump:

```
# mkfdmn /dev/disk/dsk5c  TMP_BACKUP
# mkfset TMP_BACKUP tmp_backup
# mkdir /tmp_backup
# mount TMP_BACKUP#tmp_backup /tmp_backup
# vdump -0 -f /tmp_backup/root_backup.vdump /
path     : /
dev/fset : root_domain#root
type     : advfs
advfs id : 0x3b000fb0.000919cc.1
vdump: Dumping directories
vdump: Dumping 96402959 bytes, 117 directories, 2024 files
vdump: Dumping regular files
vdump: Status at Thu May 17 12:52:38 2001
vdump: Dumped  96525730 of 96402959 bytes; 100.0% completed
vdump: Dumped  117 of 117 directories; 100.0% completed
vdump: Dumped  2024 of 2024 files; 100.0% completed
vdump: Dump completed at Thu May 17 12:52:38 2001
```

5.3.4 Recreating the root Domain on a Different Volume

To recreate the root domain on the new volume, you must restore the backup of the root domain to the new volume. This example also moves the swap partition from dsk3b to dsk6b.

1. Shut down the system booted from your old root domain.

   ```
   # shutdown -h now
   ```

2. Boot from the current operating system CD-ROM or Remote Installation Service (RIS) server. For example, from the CD-ROM:

   ```
   >>> boot dka500
   ```

   From the RIS server:

   ```
   >>> boot ewa0
   ```

3. Exit the installation.

   - If you have a VGA graphics console, choose to exit the installation, or from the File menu of the Installation and Configuration Welcome dialog box, choose shell window.
   - If you have a serial console terminal, select option `3) Exit Installation`.

   You will get a shell (#) prompt.

4. If you have backed your root domain to tape, install the tape device.

   ```
   # dn_setup -install_tape
   ```

   For more information see *System Administration* “Using dn_setup to Perform Generic Operations.”

5. Verify that the new device is recognized properly by the Operating System and that the backup device is properly installed.

   ```
   #  hwmgr -view devices
   HWID: Device Name     Mfg    Model          Location
   ------------------------------------------------------------
    4:(unknown)
    6:(unknown)
   38:/dev/disk/floppy0c    3.5in floppy     fdi0-unit-0
   41:/dev/disk/cdrom0c DEC RRD45    (C) DEC bus-0-targ-5-lun-0
   42:/dev/disk/dsk3c   DEC RZ1DB-CA (C) DEC bus-1-targ-0-lun-0
   43:/dev/disk/dsk4c   DEC RZ1CB-CA (C) DEC bus-1-targ-1-lun-0
   44:/dev/disk/dsk5c   DEC RZ1CB-CA (C) DEC bus-1-targ-2-lun-0
   45:/dev/disk/dsk6c   DEC RZ1BB-CS (C) DEC bus-1-targ-3-lun-0
   46:/dev/ntape/tape0  DEC TLZ10    (C) DEC bus-1-targ-4-lun-0
   ```

   If the new root disk Device Name is listed as `(unknown)`, check for proper hardware installation and configuration. In this example the root domain will be moved to `dsk6`. The tape backup device is `tape0` and the original root domain resides on `dsk3`.

6. Create a new root domain and root fileset on the new root device and mount it at `/mnt`.

   ```
   # mkfdmn -r /dev/disk/dsk6a root_domain
   # mkfset root_domain root
   # mount root_domain#root /mnt
   ```

7. Restore the root domain from backup.

   - If your backup is on tape:

     ```
     # vrestore -x -f /dev/tape/tape0_d1 -D /mnt
     ```

   - If your backup is on disk:

First create a directory entry for the backup domain in the /etc/fdmns directory. This new directory will only exist in the UNIX installation environment.

```
# mkdir /etc/fdmns/TMP_BACKUP
```

Then create a soft link in the new directory pointing to the volume used for the backup domain.

```
# ln -s /dev/disk/dsk5c /etc/fdmns/TMP_BACKUP/dsk5c
```

Mount the domain and fileset containing the backup. The new directory is created in /var because the installation root file system is mounted read-only.

```
# mkdir /var/tmp_backup
# mount TMP_BACKUP#tmp_backup /var/tmp_backup
```

Restore the files from the TMP_BACKUP domain to the new root domain.

```
# vrestore -x -f /var/tmp_backup/root_backup.vdump -D /mnt
vrestore: Date of the vdump save-set:Fri May 11 2001
vrestore: Save-set source directory : /
vrestore: informational: [13] posting event:
sys.unix.fs.advfs.fset.backup.lock
       If running in single user mode, EVM is not running.
       Please ignore this posting.
vrestore: informational: [13] posting event:
sys.unix.fs.advfs.fset.backup.unlock
      If running in single user mode, EVM is not running.
      Please ignore this posting.
```

The new root domain is now created and populated with files from the original root domain.

8. To finish the process, you must update system bookkeeping to point to the new root volume. In this example, the root domain and the swap partition were moved from dsk3 to dsk6. Nothing else was changed.

   Update the /etc/fdmns directory to identify the new root domain. Here dsk6a is the volume containing the new root domain and dsk3a is the volume containing the original root domain.

   ```
   # cd /mnt/etc/fdmns/root_domain
   # ln -s /dev/disk/dsk6a dsk6a
   # rm dsk3a
   ```

9. Change the swap partition in sysconfigtab in the new root domain using the editor of your choice. This example uses the vi editor.

   ```
   # vi /mnt/etc/sysconfigtab
   ```

   a. In the vm: section (stanza), change the swap device line from swapdevice=/dev/disk/dsk3b to swapdevice=/dev/disk/dsk6b. This change reflects the new location of the swap partition.

b. Save the changes and exit the editor.

10. Halt the system and change the default boot device.

```
# halt
. . .
>>> set bootdef_dev dkb300
```

11. Boot the new root domain.

```
 >>> boot
```

Retain the original root domain until you are certain that the data in the original root domain was successfully transferred to the new root domain, then remove the original domain with the rmfdmn command.

5.4 Disk File Structure Incompatibility

Domains created on operating system software Version 5.0 and later have a new on-disk format that is incompatible with earlier versions (see Section 1.6.3). The newer operating system recognizes the older disk structure, but older operating systems do not recognize the newer disk structure. If you install your new operating system software as an update to your Version 4 operating system software (not a full installation), your /root, /usr, and /var files retain a domain version number (DVN) of 3 (see Section 1.6.3.1). If you fully install your Version 5 operating system, the /root, /usr, and /var files have a DVN of 4.

To access a DVN4 fileset from an older operating system, NFS mount the fileset from a server running Version 5.0 or later operating system software, or upgrade your operating system to Version 5.0 or later.

If you try to mount a fileset belonging to a DVN4 domain when you are running a version of the operating system earlier than Version 5.0, you get an error message.

There is no tool that automatically upgrades DVN3 domains to DVN4. To upgrade a domain to DVN4, use the procedure in Section 1.6.3.2.

5.4.1 Utility Incompatibility

Because of the new on-disk file formats in Version 5.0 and later of the operating system, some AdvFS utilities from earlier releases have the potential to corrupt domains created using the new on-disk formats. All statically-linked AdvFS-specific utilities from earlier operating system versions do not run on Version 5.0 and later. These utilities are usually from operating system versions prior to Version 4.0. In addition, the following dynamically-linked AdvFS utilities from earlier releases of Tru64 UNIX do not run on Version 5.0 and later:

- `advfsstat`
- `balance`
- `chvol`
- `defragment`
- `rmvol`
- `showfdmn`
- `verify`

5.4.2 Avoiding Metadata Incompatibility

If a system crashes or goes down unexpectedly, after reboot, AdvFS performs recovery when the filesets that were mounted at the time of the crash are remounted. This recovery keeps the AdvFS metadata consistent and makes use of the AdvFS transaction log file.

Different versions of the operating system use different AdvFS log record types. Therefore, it is important that AdvFS recovery operations be done on the same version of the operating system as was running at the time of the crash.

To reboot without error using a different version of the operating system, cleanly unmount all filesets before rebooting. If the system failed due to a system panic or an AdvFS domain panic, it is best to reboot using the original version of the operating system and then run the `verify` command to ensure that the domain is not corrupted. If it is not corrupted, you can reboot your system using a different version of the operating system. If the `verify` utility indicates that the domain is corrupt, see Section 5.8.4.

5.5 Memory Mapping, Direct I/O, and Data Logging Incompatibility

Unless you have turned on atomic-write data logging by using the `mount -o adl` command, memory mapping, atomic-write data logging, and direct I/O are mutually exclusive. If a file is open in one of these modes, attempting to open the same file in a conflicting mode fails. For more information see Section 4.4, Section 4.6, and `mmap`(2).

5.6 Invalid or Corrupt Saveset Format

If you are restoring a saveset that has been written to disk and get an error message that its format is invalid or corrupt, check that you have not backed the saveset up to partition a or c, which include block 0 of the disk. Block 0, the disk label block, is protected from accidental writes to it. To dump to a partition that starts at block 0 of a disk, you must first clear the disk label. If you do not, the output from the vdump command might appear to contain valid savesets, but when the vrestore command attempts to interpret the disk label as part of the saveset, it returns an error (see Section 3.2.6).

5.7 Improving Poor Performance

The performance of a disk depends upon the I/O demands upon it. If you structure your domain so that heavy access is focused on one volume, it is likely that system performance will degrade. After you determine the load balance, there are a number of ways that you can equalize the activity and increase throughput. See *System Configuration and Tuning*, command reference pages, and Chapter 4 for more complete information.

To discover the causes of poor performance, first check system activity (see Section 4.1). There are a number of ways to improve performance:

- Upgrade domains (Section 1.6.3.2)

  DVN4 domains are indexed when a directory grows beyond a page, that is, about 200 files. Directories with more than 5000 files show the most benefit.

- Eliminate disk access incompatibility (Section 4.6)

  If you initiate direct I/O (which turns off caching) to read and write data to a file, any application that accesses the same file also has direct I/O. This might prove inefficient (see Section 4.6).

- Defragment domains (Section 4.8)

  As files grow, contiguous space on disk often is not available to accommodate new data, so files become fragmented. File fragmentation can reduce system performance because more I/O is required to read or write a file.

- Move filesets to different volumes (Section 4.11)

  You can move a domain to a volume that is larger or less congested. You can create another domain on another volume and move a fileset to it.

If you have AdvFS Utilities, you can also:

- Balance a multivolume domain (Section 4.10)

  Files that are distributed unevenly can degrade system performance. Use the balance command to redistribute the files evenly over all your volumes.

- Stripe individual files (Section 4.13)

AdvFS allows you to stripe individual files across multiple volumes. Use AdvFS striping only on directly attached storage that is not otherwise striped. Combining AdvFS striping with other striping might degrade performance.

- Migrate individual files (Section 4.12)

  You can use the `migrate` utility to move a heavily accessed file or selected pages of a file to another volume in the domain. You can move the file to a specific volume, or you can let the system choose where to move the file.

- Change AdvFS resources

  You can change your file system size in the following ways:

  - Increase the size of a domain by adding a volume using the `addvol` command (Section 1.6.6)

    For optimum performance, each volume you add should consist of the entire disk (typically, partition `c`). Do not add a volume that contains data you want to keep. When you run the `addvol` command, data on the added device is destroyed.

  - Shrink a domain by removing a volume using the `rmvol` command (Section 1.6.7)

    You can interrupt the `rmvol` process by pressing Ctrl/c without damaging your domain. Files already removed from the volume remain in their new location. Files that had not been moved at the time of the interrupt remain in their original location.

    If the volume from which the files have been removed does not allow new file allocations after an aborted `rmvol` operation, use the `chvol -A` command to reactivate the volume.

    Striped file segments are moved to a volume that does not contain a stripe. If this is not possible, the system requests confirmation before doubling up on stripes (see Section 4.13.

  - Change the size of a domain by changing volumes (Section 4.12)

    Add a new volume, move your files to it, then remove the old volume.

  See *System Limits* for the number of volumes, domains, and so forth that the AdvFS file system can handle.

5.8 Fixing Disk Problems

There are a number of problems that may be directly related to the way you are using storage.

5.8.1 Reusing Space

If you want to add storage from an existing domain (the /etc/fdmns directory entry exists) to another domain, you can remove the volume by using the rmvol command then add it to the other domain.

For example, if your volume is /dev/disk/dsk5c, your original domain is old_domain, and the domain you want to add the volume to is new_domain, mount all the filesets in old_domain, then enter:

```
# rmvol /dev/disk/dsk5c old_domain
# addvol /dev/disk/dsk5c new_domain
```

If the disk or disk partition you want to add is not part of an existing domain but is giving you a warning message because it is labeled, reset the disk label. If you answer **yes** to the prompt on the addvol or mkfdmn command, the disk label is reset. All information that is on the disk or disk partition that you are adding is lost.

5.8.2 Limiting Disk Space Usage

If your system is running without any limits on resource usage, you can add quotas to limit the amount of disk space your users can access. AdvFS quotas provide a layer of control beyond that available with UFS.

User and group quotas limit the amount of space a user or group can allocate for a fileset. Fileset quotas restrain a fileset from using all of the available space in a domain.

You can set two types of quota limits: hard limits that cannot be exceeded, and soft limits that can be exceeded for a period of time called the grace period. You can turn quota enforcement on and off. See Chapter 2 for complete information.

If you are working in an editor and realize that the information you need to save will exceed your quota limit, do not abort the editor or write the file because data might be lost. Instead, remove files to make room for the edited file before writing it. You can also write the file to another fileset, such as tmp, remove files from the fileset whose quota you exceeded, and then move the file back to that fileset.

AdvFS imposes quota limits in the rare case that you are 8 KB below the user, group, or fileset quota and are attempting to use some or all of the remaining space. This is because AdvFS allocates storage in units of 8 KB. If adding 8 KB to a file exceeds the quota limit, then that file is not extended.

5.8.3 Fixing On-Disk Metadata Corruptions

If you have a domain that cannot be mounted without a domain panic, or if the `verify` command detects on-disk corruption and is unable to fix it, run the `fixfdmn` utility. The `fixfdmn` utility is designed primarily to put a domain into a usable (mountable) state. In the process, as much data as possible is retrieved. However, if recovering data from a file is your priority, use the `salvage` utility (see Section 5.8.4).

The `fixfdmn` utility runs on unmounted filesets. It scans on-disk metadata looking for corruptions and, if enough viable data is intact, it attempts to correct the corrupt metadata. If not enough viable metadata is available, the `fixfdmn` utility attempts to bypass the corruption by moving or deleting the corrupt metadata and deleting files as necessary.

You can run the `fixfdmn -n` command to check the domain and not do any repairs.

The utility saves a message log file and two undo files. The utility can use the undo files to restore the domain to the configuration it had before you ran the `fixfdmn` command.

See `fixfdmn`(8) for more information.

5.8.4 Recovering File Data from a Corrupted Domain

The way you recover the contents of a corrupted domain depends on the nature of the corruption. Follow the recovery path for as many steps as needed. The following procedure assumes that you are only experiencing file system corruption, not hardware failure.

1. Run the `verify` command to try to repair the domain (see Section 5.2.2 and `verify`(8)). The `verify` command fixes only a limited set of problems.
2. If the `verify` command detects on-disk corruption, run the `fixfdmn` command (see Section 5.8.3 and `fixfdmn`(8)).
3. If running the `fixfdmn` command does not solve the problem, determine the date of the most recent backup. Run the `salvage` command to recover as many of the recent file changes as possible. The `salvage` command extracts salvageable files from the corrupted domain and places copies of them in filesets created to hold the recovered files. Depending on the nature of the corruption, you may be able to extract all or some of the data in the corrupted domain.

   You can use the `salvage -d` command to extract files modified after a specified date and time. If you have no backups, you can run the `salvage` utility without the `-d` option to recover all the files in the domain.

4. Recreate the domain from the latest backups then copy any files recovered with the salvage command into the recreated domain.

Use the SysMan Manage an AdvFS Domain utility, or enter the salvage command from the command line. You can recover data to disk or to tape. The amount of data you can recover depends upon the nature of the corruption to your domain. See salvage(8) for more information.

Running the salvage command does not guarantee that you will recover all files in your domain. You might be missing files, directories, file names, or parts of files. The utility generates a log file that contains the status of files that were recovered. Use the -l option to list in the log file the status of all files that are encountered.

The salvage command places the recovered files in directories named after the filesets. You can move the recovered files to new filesets. The utility creates a lost+found directory for each fileset where it puts files that have no parent directory. You can specify the pathname of the directory that is to contain the recovered fileset directories. If you do not specify a directory, the utility writes recovered filesets under the current working directory.

You can also recover data from a damaged domain to tape in a tar format.

5.8.4.1 Salvaging Data to Disk

You can recover data from a corrupted domain to another local unused disk. In this example the corrupted domain is called PERSONNEL and contains the fileset personnel_fset mounted at /personnel. The original domain is on volume /dev/disk/dsk12c and the salvage command places output on /dev/disk/dsk3c.

1. Unmount all the filesets in the corrupted domain.
2. Create a domain and a fileset to hold the recovered information and mount the fileset. For example, to mount the fileset recover_fset in the domain RECOVER mounted at /recover:

```
# mkfdmn /dev/disk/dsk3c RECOVER
# mkfset RECOVER recover_fset
# mkdir /recover
# mount RECOVER#recover_fset /recover
```

3. Run the salvage command. In this example, files from the PERSONNEL domain that were modified after 1:30 PM on December 7, 2000 are extracted from the damaged domain.

```
# /sbin/advfs/salvage -d 200012071330 -D /recover PERSONNEL
salvage: Domain to be recovered 'PERSONNEL'
salvage: Volume(s) to be used '/dev/disk/dsk12c'
salvage: Files will be restored to '/recover'
```

```
salvage: Logfile will be placed in './salvage.log'
salvage: tarting search of all filesets: 09-May-2001
salvage: tarting search of all volumes: 09-May-2001
salvage: Loading file names for all filesets: 09-May-2001
salvage: tarting recovery of all filesets: 09-May-2001
```

View the `salvage.log` file to ensure that all necessary files were recovered.

4. Recreate the domain. Here the domain is recreated on the original volume.

Caution

If you recreate a domain on the same volume as your original domain, you destroy all the data in the original domain. To save your corrupted domain, recreate the domain on a different volume.

```
# rmfdmn PERSONNEL
rmfdmn: remove domain PERSONNEL? [y/n] y
rmfdmn: informational:[13]posting event:
     sys.unix.fs.advfs.fdmn.rm
       If running in single user mode, EVM is not running
       Please ignore this posting.
rmfdmn: domain PERSONNEL removed.
# mkfdmn /dev/disk/dsk12c PERSONNEL
# mkfset PERSONNEL personnel_fset
```

5. If you are restoring some of the domain from backup, do this now. This procedure is specific to your site.

6. Copy the salvaged files from the temporary location to the restored domain and remove the recovery domain.

```
# mkdir /personnel
# mount PERSONNEL#personnel_fset /personnel
# cp -Rp /RECOVER/personnel_fset/* /personnel
# umount /recover
# rmfdmn RECOVER
rmfdmn: remove domain RECOVER [y/n] y
rmfdmn: domain RECOVER removed.
```

5.8.4.2 Salvaging Data to Tape

If your system does not have enough space to hold the information recovered by the `salvage` utility, you can recover data to tape and then write it back on to your original disk location.

To recover data from a corrupted domain called PERSONNEL on volume /dev/disk/dsk12c containing the personnel_fset fileset mounted at /personnel to tape:

1. Unmount all filesets in the corrupted domain.
2. Install a tape on the local tape drive.
3. Run the salvage command using the -F and -f options to specify tar format and the tape drive.

   In this example, files from the PERSONNEL domain that were modified after 1:30 PM on December 7, 2000 are extracted and stored on tape.

   ```
   # /sbin/advfs/salvage -d 200012071330 -F tar \
    -f /dev/tape/tape0_d1 PERSONNEL
   salvage: Domain to be recovered ’PERSONNEL’
   salvage: Volume(s) to be used ’/dev/disk/dsk12c’
   salvage: Files archived to ’/dev/tape/tape0_d1’ in TAR format
   salvage: Logfile will be placed in ’./salvage.log’
   salvage: Starting search of all filesets: 09-May-2001
   salvage: Starting search of all volumes: 09-May-2001
   salvage: Loading file names for all filesets: 09-May-2001
   salvage: Starting recovery of all filesets: 09-May-2001
   ```

   View the salvage.log file to ensure that all necessary files were recovered.
4. Recreate the domain.

   Caution

   If you recreate a domain on the same volume as your original domain, you destroy all the data in the original domain. To save your corrupted domain, recreate the domain on a new volume.

   ```
   # rmfdmn PERSONNEL
   rmfdmn: remove domain PERSONNEL? [y/n] y
   rmfdmn: informational:[13]posting event:
        sys.unix.fs.advfs.fdmn.rm
          If running in single user mode, EVM is not running
          Please ignore this posting.
   rmfdmn: domain PERSONNEL removed.
   # mkfdmn /dev/disk/dsk12c PERSONNEL
   # mkfset PERSONNEL personnel_fset
   ```
5. If you are restoring some of the domain from backup, do this now.
6. Copy the salvaged files from tape to the restored domain and remove the recovery domain.

```
# cd /
# mkdir /personnel
# mount PERSONNEL#personnel_fset /personnel
# tar -xpvf /dev/tape/tape0_d1
```

5.8.4.3 Salvaging Data from a Corrupted root Domain

If your system is not bootable because the root domain is corrupt, you can boot your system from the installation CD-ROM and run the /sbin/advfs/salvage command. Follow the steps in Section 5.11 to boot your system and exit the installation. Depending on the nature and extent of the root domain corruption, successful file recovery may not be possible.

If you are booting from the installation CD-ROM, device name assignments may differ from the assignments made on the installed operating system. Use the hwmgr -view devices command to view a table of special device names mapped to hardware identification. Be certain you are referencing the intended devices before issuing commands that destroy data.

To recover data from a corrupted root domain on volume /dev/disk/dsk0a to another local, unused disk, /dev/disk/dsk3c:

1. Create a domain and filesets to hold the recovered information and mount the filesets.

   ```
   # mkfdmn /dev/disk/dsk3c RECOVER
   # mkfset RECOVER recover_fset
   # mkdir /recover
   # mount RECOVER#recover_fset /recover
   ```

2. Run the salvage command. You must use the -V option to specify the volume that the command will operate on.

   In this example, files from the PERSONNEL domain that were modified after 1:30 PM on December 7, 2000 are extracted and stored in filesets mounted at /recover.

   ```
   # /sbin/advfs/salvage -d 200012071330 -D /recover \
   -V /dev/disk/dsk0a
   salvage: Volume(s) to be used '/dev/disk/dsk0a'
   salvage: Files will be restored to '/recover'
   salvage: Logfile will be placed in './salvage.log'
   salvage: Starting search of all filesets: 09-May-2001
   salvage: Loading file names for all filesets: 09-May-2001
   salvage: Starting recovery of all filesets: 09-May-2001
   ```

   View the salvage.log file to ensure that all necessary files were recovered.

3. Recreate the root domain as described in Section 5.11. Mount the root domain again at /mnt. If you intend to recover your root domain from backup, do so now.

4. Copy the salvaged files from the recovery location to the root domain and remove the recovery domain.

   ```
   # cd /recover
   # cp -RP * /mnt
   # cd /
   # umount /mnt /recover
   # rmfdmn RECOVER
   rmfdmn: remove domain RECOVER [y/n] y
   rmfdmn: domain RECOVER removed.
   ```

   The root domain is restored.

5.8.4.4 Salvaging Data Block by Block

If you ran the `salvage` utility and were unable to recover a large number of files, run the `salvage -S` command. This process is very slow because the utility reads every disk block at least once. If you are recovering to tape and have already created a new domain on the disks containing the corrupted domain, you cannot use the `-S` option because your original information is lost.

Note

If you have accidentally used the `mkfdmn` command on a good domain, running the `salvage -S` utility is the only way to recover files.

Caution

The `salvage` utility opens and reads block devices directly, which can present a security problem. With the `-S` option it might be possible to access data from older, deleted AdvFS domains while attempting to recover data from the current AdvFS domain.

The following example recovers data block by block.

```
# /sbin/advfs/salvage -S PERSONNEL
salvage: Domain to be recovered 'PERSONNEL'
salvage: Volume(s) to be used '/dev/disk/dsk12c'
salvage: Files will be restored to '.'
salvage: Logfile will be placed in './salvage.log'
salvage: Starting sequential search of all volumes: 09-May-2001
salvage: Loading file names for all filesets: 09-May-2001
salvage: Starting recovery of all filesets: 09-May-2001
```

5.8.5 “Can’t Clear a Bit Twice” Error Message

If you receive a "Cannot clear a bit twice" error message, your domain is damaged. To repair it:

1. Set the `AdvfsFixUpSBM` kernel variable to allow access to the damaged domain. This flag is off by default. To turn it on:

   ```
   # dbx -k /vmunix /dev/mem
   dbx> assign AdvfsFixUpSBM = 1
   dbx> quit
   ```

2. Mount and back up the filesets in the damaged domain.
3. Turn `AdvfsFixUpSBM` off:

   ```
   # dbx -k /vmunix /dev/mem
   dbx> assign AdvfsFixUpSBM = 0
   dbx>  quit
   ```

4. Unmount the filesets in the domain. Run the `verify -f` utility. If there are errors, continue through steps 5 and 6.
5. Recreate the domain and filesets.
6. Restore from the backup.

Note

The `AdvfsFixUpSBM` variable is global. Turn it off so that the error message is again available for all domains.

5.8.6 Recovering from a Domain Panic

When a metadata write error occurs, or if corruption is detected in a single AdvFS domain, the system initiates a **domain panic** (rather than a system panic) on the domain. This isolates the failed domain and allows a system to continue to serve all other domains. After a domain panic, AdvFS no longer issues I/O requests to the disk controller for the affected domain. Although the domain cannot be accessed, the filesets in the domain can be unmounted.

When a domain panic occurs, an `EVM` event is logged (see `EVM`(5)) and the following message is printed to the system log and the console:

`AdvFS Domain Panic; Domain` *`name`* **`Id`** *`domain_Id`*

For example:

```
AdvFS Domain Panic; Domain staffb_domain Id 2dad7c28.0000dfbb
An AdvFS domain panic has occurred due to either a
```

```
 metadata write error or an internal inconsistency.
This domain is being rendered inaccessible.
```

By default, a domain panic on an active domain causes a live dump to be created and placed in the `/var/adm/crash` directory. Some AdvFS-related errors might also be recorded in `/var/adm/binary.errlog`. Please file a problem report with your software support organization and include the dump file and a copy of the running kernel.

To recover from a domain panic, perform the following steps:

1. Run the `mount -t` command and identify all mounted filesets in the affected domain.
2. Unmount all the filesets in the affected domain.
3. Examine the `/etc/fdmns` directory to obtain a list of the AdvFS volumes in the domain that panicked.
4. Run the `savemeta` command (see `savemeta`(8)) to collect information about the metadata files for each volume in the domain. Technical support needs this information.
5. If the problem is a hardware problem, fix it before continuing.
6. Run the `verify` utility on the domain (see Section 5.2.2).
    - If there are no errors, mount all the filesets you unmounted and resume normal operations.
    - If the `verify` command runs but shows errors, mount the filesets, do a backup, and recreate the domain. Note that the backup might be incomplete and that earlier backup resources might be needed.
7. If the failure prevents complete recovery, recreate the domain on new volumes by using the `mkfdmn` command and restore the domain's data from backup. If the backup does not provide enough information, you might need to run the `salvage` utility (see Section 5.8.4).

For example:

```
# mount -t advfs
staffb_dmn#staff3_fs on /usr/staff3 type advfs (rw)
staffb_dmn#staff4_fs on /usr/staff4 type advfs (rw)
# umount /usr/staff3
# umount /usr/staff4
# ls -l /etc/fdmns/staffb_dmn
lrwxr-xr-x 1 root system 10 Nov 04 16:46
    dsk35c->/dev/disk/dsk3c
lrwxr-xr-x 1 root system 10 Nov 04 16:50
    dsk36c->/dev/disk/dsk6c
lrwxr-xr-x 1 root system 10 Nov 04 17:00
    dsk37c->/dev/disk/dsk1c
```

```
# savemeta staffb_dmn /tmp/saved_dmn
# verify staffb_dmn
```

You do not need to reboot after a domain panic.

If you have recurring domain panics, you might try adjusting the AdvfsDomainPanicLevel attribute (see Section 4.14) in order to facilitate debugging.

5.8.7 Recovering from Filesets That are Mounted Read-Only

When a fileset is mounted, AdvFS verifies that all volumes in a domain can be accessed. The size recorded in the domain's metadata for each volume must match the size of the volume. If the sizes match, the mount proceeds. If a volume is smaller than the recorded size, AdvFS attempts to read the last block marked in use for the fileset. If this block can be read, the mount succeeds, but the fileset is marked as read-only. If the last in-use block for any volume in the domain cannot be read, the mount fails. See mount(8) for more information.

If a fileset is mounted read-only, check the labels of the flagged volumes in the error message. There are two common errors:

- A disk is mislabeled on a RAID array.
- An LSM volume upon which an AdvFS domain resides was shrunk from its original size (see Section 1.10).

If you have AdvFS Utilities, and if the domain consists of multiple volumes with enough free space to remove the offending volume, you do not need to remove your filesets. However, you should back them up before proceeding.

1. Remove the volume from the domain by using the rmvol command. (This automatically migrates the data to the remaining volumes.)
2. Correct the disk label of the volume by using the disklabel command.
3. Add the corrected volume back to the domain by using the addvol command.
4. Run the balance command to distribute the data across the new volumes.

For example, if /dev/disk/dsk2c (on a device here called <disk>) within the data5 domain is mislabeled, you can migrate your files on that volume (automatic with the rmvol command), then move them back after you restore the volume.

```
# rmvol /dev/disk/dsk2c data5
# disklabel -z dsk2
# disklabel -rw dsk2 <disk>
# addvol /dev/disk/dsk2c data5
# balance data5
```

If you do not have AdvFS Utilities, or if there is not enough free space in the domain to transfer the data from the offending volume:

1. Back up all filesets in the domain.
2. Remove the domain by using the `rmfdmn` command.
3. Correct the disk label of the volume by using the `disklabel` command.
4. Make the new domain.
5. If you have AdvFS Utilities and if the original domain was multivolume, add the corrected volume back to the domain by using the `addvol` command.
6. Restore the filesets from the backup.

For example, if `/dev/disk/dsk1c` (on a device here called <disk>) containing the `data3` domain is mislabeled:

```
# vdump -0f -u /data3
# rmfdmn data3
# disklabel -z dsk1 <disk>
# disklabel -w dsk1 <disk>
# mkfdmn data3
```

If you are recreating a multivolume domain, include the necessary `addvol` commands to add the additional volumes. For example to add `/dev/disk/dsk5c` to the domain:

```
# addvol /dev/disk/dsk5c data3
# mkfset data3 data3fset
# mount data3#data3fset /data3
# vrestore -xf - /data3
```

5.9 Restoring the /etc/fdmns Directory

AdvFS must have a current `/etc/fdmns` directory in order to mount filesets (see Section 1.6.2). A missing or damaged `/etc/fdmns` directory prevents access to a domain, but the data within the domain remains intact. You can restore the `/etc/fdmns` directory from backup or you can recreate it.

It is preferable to restore the `/etc/fdmns` directory from backup if you have a current backup copy. You can use any standard backup facility (`vdump`, `tar`, or `cpio`) to back up the `/etc/fdmns` directory. To restore the directory, use the recovery procedure that is compatible with your backup process.

If you cannot restore the `/etc/fdmns` directory, you can reconstruct it manually (see Section 5.9.1) or with the `advscan` command (see Section 5.9.2). The procedure for reconstructing the `/etc/fdmns` directory is similar for both single-volume and multivolume domains. You can construct the directory for a missing domain, missing links, or the whole directory.

If you choose to reconstruct the directory manually, you must know the name of each domain and its associated volumes.

5.9.1 Reconstructing the /etc/fdmns Directory Manually

If you accidentally lose all or part of your `/etc/fdmns` directory, and you know which domains and links are missing, you can reconstruct it manually.

The following example reconstructs the `/etc/fdmns` directory and two domains, In this example the domains exist and their names are known. Each domain contains a single volume (or special device). Note that the order of creating the links in these examples does not matter. The domains are:

`domain1` on `/dev/disk/dsk1c`

`domain2` on `/dev/disk/dsk2c`

To reconstruct the two single-volume domains, enter:

```
# mkdir /etc/fdmns
# mkdir /etc/fdmns/domain1
# cd /etc/fdmns/domain1
# ln -s /dev/disk/dsk1c dsk1c
# mkdir /etc/fdmns/domain2
# cd /etc/fdmns/domain2
# ln -s /dev/disk/dsk2c dsk2c
```

The following example reconstructs one multivolume domain. The `domain1` domain contains the following three volumes:

`/dev/disk/dsk1c`

`/dev/disk/dsk2c`

`/dev/disk/dsk3c`

To reconstruct the multivolume domain, enter:

```
# mkdir /etc/fdmns
# mkdir /etc/fdmns/domain1
# cd /etc/fdmns/domain1
# ln -s /dev/disk/dsk1c dsk1c
# ln -s /dev/disk/dsk2c dsk2c
# ln -s /dev/disk/dsk3c dsk3c
```

5.9.2 Reconstructing the /etc/fdmns Directory Using advscan

You can use the `advscan` command to determine which partitions on a disk or which Logical Storage Manager (LSM) volumes are part of an AdvFS domain. Then you can use the command to rebuild all or part of your /etc/fdmns directory. This command is useful:

- If you moved disks to a new system, if device numbers have changed, or if you lost track of a domain location
- For repair, if you delete the /etc/fdmns directory, delete a domain from the /etc/fdmns directory, or delete links from a domain's subdirectory in the /etc/fdmns directory

The advscan command can:

- Determine if a partition is an AdvFS partition.
- List partitions in the order they are found on disk.
- Read the disk label to determine which partitions are in the domain and if any are overlapping.
- Scan all disks found in any /etc/fdmns domain.
- Recreate missing domain directories. The domain name is created from the device name.
- Fix the domain count and links for a domain.

For each domain there are three numbers that must match for the AdvFS file system to operate properly:

- The number of physical partitions found by the advscan command that have the same domain ID
- The domain volume count (the number stored in the AdvFS metadata that specifies the number of partitions in the domain)
- The number of /etc/fdmns links to the partitions, because each partition must be represented by a link

See advscan(8) for more information.

Inconsistencies can occur in these numbers for several reasons. In general, the advscan command treats the domain volume count as more reliable than the number of partitions or the /etc/fdmns links. The following tables list anomalies, possible causes, and corrective actions that the advscan utility can take. In the table, the letter N represents the value that is expected to be consistent for the number of partitions, the domain volume count, and the number of links.

Table 5–2 shows possible causes and corrective actions if the expected value, N, for the number of partitions and for the domain value count do not equal the number of links in the /etc/fdmns/<dmn> directory.

Table 5–2: Fileset Anomalies and Corrections - Links Not Equal

Number of Links in /etc/fdmns/ <dmn>	Possible Cause	Corrective Action
<N	addvol terminated early or a link in /etc/fdmns/<dmn> was manually removed.	If the domain is activated before running the advscan -f command and the cause of the mismatch is an interrupted addvol command, the situation is corrected automatically. Otherwise, advscan utility adds the partition to the /etc/fdmns/<dmn> directory.
>N	rmvol terminated early or a link in /etc/fdmns/<dmn> was manually added.	If the domain is activated and the cause of the mismatch is an interrupted rmvol command, the situation is corrected automatically. If the cause Is a manually added link in /etc/fdmns/<dmn>, systematically try removing different links in the /etc/fdmns/<dmn> directory and activating the domain. The number of links to remove is the number of links in the /etc/fdmns/<dmn> directory minus the domain volume count displayed by advscan.

Table 5–3 shows possible causes and corrective actions if the expected value, N, for the number of partitions and for the number of links in the /etc/fdmns/<dmn> directory do not equal the domain volume count.

Table 5–3: Fileset Anomalies and Corrections - Domain Volume Count Not Equal

Domain Volume Count	Possible Cause	Corrective Action
<N	Cause unknown.	Cannot correct; run the salvage utility to recover as much data as possible from the domain.
>N	The addvol command terminated early and the partition being added is missing or was reused.	Cannot correct; run the salvage utility to recover as much data as possible from the remaining volumes in the domain.

Table 5–4 shows possible causes and corrective actions if the expected value, N, for the domain volume count and for the number of links in the /etc/fdmns/<dmn> directory do not equal the number of partitions.

Table 5–4: Fileset Anomalies and Corrections - Number of Partitions Not Equal

Number of Partitions	Possible Cause	Corrective Action
<N	Partition missing.	Cannot correct; run the salvage utility to recover as much data as possible from the remaining volumes in the domain.
>N	The addvol command terminated early.	None; domain mounts with N volumes; rerun the addvol command.

In the following example no domains are missing. The advscan command scans devices dsk0 and dsk5 for AdvFS partitions and finds nothing amiss. Two partitions are found, dsk0c and dsk5c, the domain volume count reports two, and two links are entered in the /etc/fdmns directory.

```
# advscan dsk0 dsk5
Scanning disks  dsk0 dsk5
Found domains:
usr_domain
                Domain Id       2e09be37.0002eb40
                Created         Thu Feb 24 09:54:15 2000
                Domain volumes          2
                /etc/fdmns links        2
                Actual partitions found:
                                        dsk0c
                                        dsk5c
```

In the following example, directories that define the domains that include dsk6 were removed from the /etc/fdmns directory. This means that the number of /etc/fdmns links, the number of partitions, and the domain volume counts are no longer equal. In this example the advscan command scans device dsk6 and recreates the missing domains as follows:

1. A partition is found containing an AdvFS domain. The domain volume count reports one, but there is no domain directory in the /etc/fdmns directory that contains this partition.
2. Another partition is found containing a different AdvFS domain. The domain volume count is also one. There is no domain directory that contains this partition.
3. No other AdvFS partitions are found. The domain volume counts and the number of partitions found match for the two discovered domains.
4. The advscan command creates directories for the two domains in the /etc/fdmns directory.
5. The advscan command creates symbolic links for the devices in the /etc/fdmns domain directories.

The command and output are as follows:

```
# advscan -r dsk6
Scanning disks  dsk6
Found domains:
*unknown*
                Domain Id       2f2421ba.0008c1c0
                Created         Thu Jan 20 13:38:02 2000

                Domain volumes          1
                /etc/fdmns links        0

                Actual partitions found:
                                        dsk6a*

*unknown*
                Domain Id       2f535f8c.000b6860
                Created         Fri Feb 25 09:38:20 2000

               Domain volumes          1
                /etc/fdmns links       0

                Actual partitions found:
                                        dsk6b*
Creating /etc/fdmns/domain_dsk6a/
        linking dsk6a

Creating /etc/fdmns/domain_dsk6b/
        linking dsk6b
```

5.10 Recovering from Corruption of a Domain

Some problems occur in AdvFS because of hardware errors. For example, if a write to the file system fails due to a hardware fault, it might appear as metadata corruption. Hardware problems cannot be repaired by your file system.

If unexplained errors on a volume in a multivolume domain, do the following:

1. As root user, examine the `/var/adm/messages` file for AdvFS I/O error messages. For example:

   ```
   Dec 05 15:39:16 systemname vmunix: AdvFS I/O error:
   Dec 05 15:39:16 systemname vmunix: Domain#Fileset:test1#tstfs
   Dec 05 15:39:16 systemname vmunix: Mounted on: /test1
   Dec 05 15:39:17 systemname vmunix: Volume: /dev/rz11c
   Dec 05 15:39:17 systemname vmunix: Tag: 0x00000006.8001
   Dec 05 15:39:17 systemname vmunix: Page: 76926
   Dec 05 15:39:17 systemname vmunix: Block: 5164080
   Dec 05 15:39:17 systemname vmunix: Block count: 256
   Dec 05 15:39:17 systemname vmunix: Type of operation: Read
   Dec 05 15:39:17 systemname vmunix: Error: 5
   Dec 05 15:39:17 systemname vmunix: To obtain the name of
   Dec 05 15:39:17 systemname vmunix: the file on which the
   Dec 05 15:39:17 systemname vmunix: error occurred, type the
   Dec 05 15:39:17 systemname vmunix: command
   Dec 05 15:39:17 systemname vmunix: /sbin/advfs/tag2name
   Dec 05 15:39:17 systemname vmunix: /test1/.tags/6
   ```

   This error message describes the domain, fileset, and volume on which the error occurred. It also describes how to find out which file was affected by the I/O error. If you have no AdvFS I/O error messages but still have unexplained behavior on the file system, unmount the domain as soon as possible and run the `verify` utility (see Section 5.2.2) to check the consistency of the domain's metadata.

2. Check for device driver error messages for the volume described in the AdvFS I/O error message. If you have no error messages, unmount the domain as soon as possible and run the `verify` utility to check the integrity of the domain's metadata. If there are no device driver I/O error messages that correspond to the AdvFS I/O error messages, then the file system is being affected by problems with the underlying hardware.

3. Try to remove the faulty volume by using the `rmvol` utility (see Section 1.6.7). If this succeeds, the file system problems should not recur.

If rmvol fails due to more I/O errors, you must recreate the domain.

a. If you have a recent backup, recreate the domain and restore it from backup. If you have no backup, or if it is too old, use the salvage utility (see Section 5.8.4) to extract the contents of the corrupted domain.

b. Remove the faulty domain by using the rmfdmn command.

c. Recreate the domain by using the mkfdmn command. Remember that if you are recreating your domain, it will have a DVN of 4 by default (see Section 1.6.3). Add volumes as needed if you have the AdvFS Utilities license. Do not to include the faulty volume in the new domain.

d. Restore the contents of the recreated domain using the information obtained in the backup step.

e. Remount the filesets in the domain.

5.11 Recovering from Corruption of an AdvFS root Domain

Catastrophic corruption of your AdvFS root domain typically requires that you recreate your root file system in order to have a bootable system. This section explains recovering a corrupted root domain on a non-clustered system. For other configurations, see *System Administration* “Duplicating or Recovering a System (Root) Disk” and *Cluster Administration*. If your root volume is an LSM volume, see *Logical Storage Manager*.

Follow this procedure if the root domain is corrupt. This procedure assumes that the hardware disk device containing the corrupted root domain is functioning properly, that the disklabel is correct, and that the problem is due to data corruption. You must be root user to reconstruct the root domain.

Depending on your system configuration, you might need the following:

- Information about console commands

  You will use Alpha System Reference Manual (SRM) console commands at the system console prompt (>>>) to perform some tasks. These commands are documented in the hardware manual for your Alpha system. If you cannot find the printed document, it is usually shipped as a printable file on a CD-ROM supplied with the system.

- A current operating system CD-ROM

  You can use the operating system CD-ROM that is packaged with the distribution media to boot your system and perform maintenance activities on various utilities.

If your local site provides a Remote Installation Service (RIS) server, you can boot your system across the network. If you choose RIS services, follow your site-specific procedures and consult the *Installation Guide*.

- Recent root domain backup media (full and recent incremental backups)

  You will need to recreate the root domain on the boot device. You are best prepared if you have a full and recent backup of the root domain. If you do not have adequate backup, depending on the nature and extent of the root domain corruption, you may be able to recover root files using the `salvage` utility. The `salvage` utility may also be used to recover files that were modified or created following the most recent backup.

5.11.1 Identifying the Hardware Resources

You need to identify the following hardware resources to complete the restoration of your root disk.

5.11.1.1 SRM Console Names for CD-ROM Drive or Network Interface Device

If you plan to boot your system from the operating system CD-ROM, determine the name of your CD-ROM drive. One method of identifying your CD-ROM drive is by issuing the `show device` command at the SRM console prompt.

```
  >>> show device | grep -E 'RR|CD'
DKA400        RRD47   1206   dka400.4.0.5.0
```

In this example, the CD-ROM device name is `DKA400` according to the SRM console firmware.

If you plan to boot your system from a RIS server, determine the name of your network interface device. One method of identifying your network interface device is by issuing the `show device` command at the SRM console prompt.

```
>>> show device | more
....
ewa0.0.0.8.0    EWA0     08-00-2B-C3-E3-DC
...
```

In this example, the network interface device name is `EWA0` according to the SRM console firmware.

For additional information, see the hardware manual for your system. For information about RIS servers, see the *Installation Guide — Advanced Topics*.

5.11.1.2 SRM Console Boot Device Name

In previous versions of the operating system, device names were assigned based on the physical location of the drive on an I/O bus. In Version 5.0 and later operating system software, device names are assigned logically and stored in a database. These names are independent of the device's physical location.

You must determine the boot device name according to the SRM console. If your boot device is the default boot device, you can identify this device using the `show bootdef_dev` command at the SRM console prompt.

```
>>> show bootdef_dev
bootdef_dev      dkb400.4.0.5.1
```

If your boot device is not the default boot device, use the `show device` command from the SRM console prompt to identify your boot device from the list.

For example, if `dkb400` is the boot device, `dk` indicates that the device is a SCSI disk, the `b` indicates that the device is connected to SCSI bus `b`, and the `400` indicates that the device's SCSI target ID is `4` and its logical unit number (LUN) is `00`. Thus, in this example, the bus/target/LUN information is `1/4/00`. This information identifies the device when you restore your domain.

5.11.1.3 UNIX Device Names

If the root domain is mountable when you boot from the installation media, the installation procedure attempts to read the existing device database from the installed root domain. If this read succeeds, the following message appears on the console:

```
Attempting to mount previous root file system disk
to save hardware configuration information...
done
```

If the hardware database read fails, messages similar to the following appears on the console:

```
Attempting to mount previous root file system disk
to save hardware configuration information...
FAILED

Unable to retain old hardware configuration from
     SCSI 1 4 0 0 0 6000 10201 077

Unable to save existing hardware configuration.
New configuration will be used.
```

If the hardware database read fails, you must translate the UNIX device name assignments to the proper hardware device by identifying the device by its bus/target/LUN (see Section 5.11.2).

5.11.2 Applying the Procedure

The following steps recover your failed root domain.

1. Boot the system using one of the following methods:

   - Insert and boot your installation CD-ROM using the device name that you determined previously. For example:

     ```
     >>> boot dka400
     ```

   - Boot from your local RIS server. For example:

     ```
     >>> boot ewa0
     ```

2. Exit the installation as follows:

   - If you have a VGA graphics console, choose to exit the installation, or from the File menu of the Installation and Configuration Welcome dialog box, choose shell window.
   - If you have a serial console terminal, select option `3) Exit Installation`.

   You will get a shell (#) prompt.

3. Identify both the bus/target/LUN of the target disk that will be used as the restored root disk and the status of backup device by using the `hwmgr -view devices` command.

   ```
   # hwmgr -view devices
   HWID: Device Name     Mfg Model         Location
   ------------------------------------------------------------
   38:/dev/disk/floppy0c    3.5in floppy     fdi0-unit-0
   41:/dev/disk/dsk0c   DEC RZ1DB-CA (C) DEC bus-1-targ-4-lun-0
   42:/dev/disk/dsk1c   DEC RZ1CB-CA (C) DEC bus-1-targ-5-lun-0
   43:/dev/disk/dsk2c   DEC RZ1CB-CA (C) DEC bus-1-targ-6-lun-0
   44:/dev/disk/cdrom0  DEC RRD47    (C) DEC bus-0-targ-5-lun-0
   47:(unknown)         DEC TLZ10    (C) DEC
   bus-1-targ-4-lun-0
   ```

   In this example, the SRM console is identified `DKB400` and the disk located at bus `b`, target `4`, LUN `0`. According to the hardware database, this same disk is identified as `dsk0` (see Section 5.11.1.2). In this procedure, `/dev/disk/dsk0a` will be used as the volume containing the corrupted root domain. A new root domain will be created on `/dev/disk/dsk0a` and files from the old root domain will be restored on it.

   To visually confirm that you have identified the correct device, use the `hwmgr —flash` command to cause the disk's light to flash for thirty seconds.

   ```
   # /sbin/hwmgr -flash light -dsf /dev/disk/dsk0a
   ```

If you plan to recover from a local tape device, identify the device in the list displayed by the `hwmgr` utility. If you do not see the tape device, check for proper installation and hardware configuration.

4. If you have a tape backup device, install it.

   ```
   # dn_setup -install_tape
   ```

   For more information see *System Administration* “Using dn_setup to Perform Generic Operations.”

   To verify the installation, repeat the `hwmgr` command.

5. If necessary, recover files with the `salvage` command and save them to a temporary domain (see Section 5.8.4).

6. Create the new root domain and root fileset. Mount the fileset at `/var/mnt`.

   ```
   # mkfdmn -r /dev/disk/dsk0a root_domain
   Warning: /dev/disk/dsk0a is marked in use for AdvFS.
   If you continue with the operation you can
   possibly destroy existing data.
   CONTINUE? [y/n] y
   # mkfset root_domain root
   # mkdir /var/mnt
   # mount root_domain#root /var/mnt
   ```

7. Use the `vrestore` command to restore the files from backup device you installed earlier.

   ```
    # vrestore -xf  /dev/tape/tape0 -D /var/mnt
   ```

8. If necessary, copy files recovered with the `salvage` command into the newly created root domain (see Section 5.8.4).

9. Halt the system.

   ```
   # halt
   ```

10. Boot the system.

    ```
    >>> boot
    ```

11. Verify success by checking the boot process for error messages.

12. It is a good idea to use the `dsfmgr` command to verify and fix the device databases and device special file names. For example:

    ```
    # dsfmgr -v
    ```

If the procedure was not successful and hardware failures are not present, your only recourse is to reinstall the operating system from the distribution media and recreate your customized environment from backup media.

5.12 Restoring a Multivolume usr Domain

Before you restore a multivolume /usr file system, you must first reconstruct the usr_domain domain with all of its volumes. However, restoring a multivolume domain requires the License Management Facility (LMF). LMF controls AdvFS Utilities, which includes the addvol command needed for creating multivolume domains.

First create a one volume usr domain and restore the addvol command. Then restore LMF and use it to enable the addvol command. When this is complete, you can add volumes to the usr domain and restore the complete multivolume domain.

LMF has two parts. A utility is stored in /usr/sbin/lmf and a database is stored in /var/adm/lmf. On some systems /var is a link to /usr and both directories are located in the usr fileset. If your system has this configuration, recover the addvol command and recover both parts of the LMF. On systems where the /usr and /var directories are located in separate filesets in usr_domain, recover the addvol command and the LMF utility into the usr fileset and recover the LMF database into the var fileset.

The following example shows how to restore a multivolume domain where the /var directory and the /usr directory are both in the usr fileset in usr_domain. The domain consists of the dsk1g, dsk2c, and dsk3c volumes. The procedure assumes that the root file system has already been restored. If it has not, see Section 5.11.

1. Mount the root fileset as read/write.

   ```
   # mount -u /
   ```

2. Remove the links for the old usr_domain and create a new usr_domain using the initial volume.

   ```
   # rm -rf /etc/fdmns/usr_domain
   # mkfdmn /dev/disk/dsk1g usr_domain
   ```

3. Create and mount the /usr and /var filesets.

   ```
   # mkfset usr_domain usr
   # mount -t advfs usr_domain#usr /usr
   ```

4. Create a soft link in /usr because that is where the lmf command looks for its database.

   ```
   # ln -s /var /usr/var
   ```

5. Insert the /usr backup tape.

   ```
   # cd /usr
   # vrestore -vi
   (/) add sbin/addvol
   (/) add sbin/lmf
   ```

```
(/) add var/adm/lmf
(/) extract
(/) quit
```

6. Reset the license database.

   ```
   # /usr/sbin/lmf reset
   ```

7. Add the extra volumes to `usr_domain`.

   ```
   # /usr/sbin/addvol /dev/disk/dsk2c usr_domain
   # /usr/sbin/addvol /dev/disk/dsk3c usr_domain
   ```

8. Do a full restore of the `/usr` backup.

   ```
   # cd /usr
   # vrestore -xv
   ```

The following example shows how to restore a multivolume domain where the `/usr` and `/var` directories are in separate filesets in the same multivolume domain, `usr_domain`. The domain consists of the `dsk1g`, `dsk2c`, and `dsk3c` volumes. In this case you must mount both the `/var` and the `/usr` backup tapes. The procedure assumes that the root file system has already been restored. If it has not, see Section 5.11.

1. Mount the root fileset as read/write.

   ```
   # mount -u /
   ```

2. Remove the links for the old `usr_domain` and create a new `usr_domain` using the initial volume.

   ```
   # rm -rf /etc/fdmns/usr_domain
   # mkfdmn /dev/disk/dsk1g usr_domain
   ```

3. Create and mount the `/usr` and `/var` filesets.

   ```
   # mkfset usr_domain usr
   # mkfset usr_domain var
   # mount -t advfs usr_domain#usr /usr
   # mount -t advfs usr_domain#var /var
   ```

4. Insert the `/var` backup tape and restore from it.

   ```
   # cd /var
   # vrestore -vi
   (/) add adm/lmf
   (/) extract
   (/) quit
   ```

5. Insert the `/usr` backup tape.

   ```
   # cd /usr
   # vrestore -vi
   (/) add sbin/addvol
   (/) add sbin/lmf
   ```

```
(/) extract
(/) quit
```

6. Reset the license database.

```
# /usr/sbin/lmf reset
```

7. Add the extra volumes to usr_domain.

```
# /usr/sbin/addvol /dev/disk/dsk2c usr_domain
# /usr/sbin/addvol /dev/disk/dsk3c usr_domain
```

8. Do a full restore of /usr backup.

```
# cd /usr
# vrestore -xv
```

9. Insert the /var backup tape and do a full restore of /var backup.

```
# cd /var
# vrestore -xv
```

5.13 Recovering from a System Crash

When each domain is mounted after a crash, the system automatically runs recovery code that checks the transaction log file to ensure that file system operations that were occurring when the system crashed are either completed or backed out. This ensures that AdvFS metadata is in a consistent state after a crash. If you are recovering your system by using an operating system other than the one that crashed, see Section 5.4.

5.13.1 Saving Copies of System Metadata

If it appears that a domain is corrupted or it is otherwise causing problems, run the savemeta command to save a copy of the domain's metadata for examination by support personnel. You must be root user to run this command (see savemeta(8)).

5.13.2 Physically Moving an AdvFS Disk

If a machine has failed, you can move disks containing AdvFS domains to another computer running the AdvFS software. Connect the disk(s) to the new machine and modify the /etc/fdmns directory so the new system recognizes the transferred volume(s). You must be root user to complete this process.

You cannot move DVN4 domains to systems running Version 4 of the operating system software. Doing so generates an error message (see Section 5.4). You can move DVN3 domains from a Version 4 machine to a machine running Version 5. The newer operating system recognizes the domains created earlier.

Caution

Do not use either the addvol command or the mkfdmn command to add the volumes to the new machine. Doing so will delete all data on the disk you are moving. See Section 5.8.4 if you have already done so.

If you do not know which partitions your domains were on, you can add the disks on the new machine and run the advscan utility, which might be able to recreate this information. You can also look at the disk label on the disk to see which partitions in the past were made into AdvFS partitions. The disk labels do not tell you which partitions belong to which domains.

If the motherboard of your machine fails, you must move the disks to another system. You might need to reassign the disk SCSI IDs to avoid conflicts. (See your disk manufacturer instructions for more information.)

For example, assume the IDs are assigned to disks 6 and disk 8. Assume also that the system has a domain, testing_domain, on two disks, dsk3 and dsk4. This domain contains two filesets: sample1_fset and sample2_fset. These filesets are mounted on /data/sample1 and /data/sample2. Assume you know that the domain that you are moving had partitions dsk3c, dsk4a, dsk4b, and dsk4g. Take the following steps to move the disks:

1. Shut down the working machine to which you are moving the disks.
2. Connect the disks from the bad machine to the good one.
3. Reboot. You do not need to reboot to single-user mode; multiuser mode works because you can complete the following steps while the system is running.
4. Determine the device nodes created for the new disks.

   ```
   # /sbin/hwmgr -show scsi -full
   ```

   The output is a detailed list of information about all the disks on your machine. The DEVICE FILE column shows the name that the system uses to refer to each disk. Find the listings for the disks that you just added, for example, disk6 anddisk8. Use these names to set up symbolic links in step 5.
5. Modify your /etc/fdmns directory to include the information from the transferred domains.

   ```
   # mkdir -p /etc/fdmns/testing_domain
   # cd /etc/fdmns/testing_domain
   # ln -s /dev/disk/dsk6c dsk6c
   # ln -s /dev/disk/dsk8a dsk8a
   # ln -s /dev/disk/dsk8b dsk8b
   ```

```
# ln -s /dev/disk/dsk8g dsk8g
# mkdir /data/sample1
# mkdir /data/sample2
```

6. Edit the /etc/fstab file to add the fileset mount-point information.

   ```
   testing_domain#sample1_fset /data/sample1 advfs rw 1 0
   testing_domain#sample2_fset /data/sample2 advfs rw 1 0
   ```

7. Mount the volumes.

   ```
   # mount /data/sample1
   # mount /data/sample2
   ```

   Note that if you run the mkfdmn command or the addvol command on partition dsk6c, dsk8a, dsk8b, dsk8g, or an overlapping partition, you will destroy the data on the disk. See Section 5.8.4 if you have accidentally done so.

5.13.3 Log File Inconsistency

When a system crashes, AdvFS performs recovery at reboot. Filesets that were mounted at the time of the crash are recovered when they are remounted. This recovery keeps the AdvFS metadata consistent and makes use of the AdvFS transaction log file.

Since different versions of the operating system use different transaction log file structures, it is important that you recover your filesets on the version of the operating system that was running at the time of the crash. If you do not, you risk corrupting the domain metadata and/or panicking the domain.

If the system crashed because you set the AdvfsDomainPanicLevel attribute (see Section 4.7) to promote a domain panic to a system panic, run the verify command on the panicked domain to ensure that it is not damaged. If your filesets were unmounted at the time of the crash, or if you remounted them successfully and ran the verify command (if needed), you can mount the filesets on a different version of the operating system, if appropriate.

5.13.4 Recovering from Problems Removing Volumes

If the removal process is interrupted (see Section 1.6.7), under some circumstances the volume can be left in an inaccessible state where you cannot write to it. These volumes are marked as “data unavailable” in the output of the showfdmn command. If the volume does not allow writes after an aborted rmvol operation, use the chvol -A command to reactivate the volume.

A

Accessing File System Information with the SysMan Menu

You must be root to use the SysMan Menu. You can open the menu:

- From the command line by entering:

  ```
  # /usr/sbin/sysman
  ```

- From the Common Desktop Environment (CDE) front panel if your system is running in a graphics environment with CDE:

  1. Select the Application Manager from the front panel.
  2. Select the System_Admin group.
  3. Select the SysMan Menu group.
  4. Select the appropriate utility from the SysMan Menu group.

For AdvFS tasks select the Storage management task then choose File System Management Utilities. The Advanced File System (AdvFS) Utilities option lists the AdvFS-related tasks.

See `sysman`(8) for more information.

B
AdvFS Commands

This appendix summarizes the AdvFS commands. The reference pages provide detailed information.

B.1 AdvFS Base System Commands

The following tables list and describe each of the AdvFS commands available in the base portion of AdvFS. These commands are included with the basic license; they do not require a layered product license. If you installed the reference page subset, you can access reference pages for each of these commands by issuing the `man` command.

Table B–1: AdvFS Configuration Commands

Command	Description
`chfile`	Changes the attributes of a file
`chvol`	Changes the attributes of a volume
`defragment`	Makes the files in a domain more contiguous
`mkfdmn`	Creates a domain
`mkfset`	Creates a fileset within a domain
`renamefset`	Renames an existing fileset
`rmfdmn`	Removes a domain
`rmfset`	Removes a fileset from a domain
`switchlog`	Moves the AdvFS log file to a different volume in a domain

Table B–2: AdvFS Information Display Commands

Command	Description
`advfsstat`	Displays file system statistics
`ncheck`	Displays the tag and full pathname for each file in the file system
`showfdmn`	Displays the attributes of a domain
`showfile`	Displays the attributes of a file

Table B–2: AdvFS Information Display Commands (cont.)

Command	Description
`showfsets`	Displays the attributes of filesets in a domain
`vdf`	Displays used and available disk space for a fileset or a domain

Table B–3: AdvFS Backup Commands

Command	Description
`rvdump`	Remotely performs full and incremental fileset backup
`rvrestore`	Remotely restores files from backup media
`vdump`	Performs full and incremental fileset backup
`vrestore`	Restores files from backup media

Table B–4: AdvFS Check and Repair Commands

Command	Description
`advscan`	Locates AdvFS partitions on disks
`fixfdmn`	Fixes on-disk metadata corruption
`mountlist`	Checks for mounted AdvFS filesets
`salvage`	Recovers file data from damaged AdvFS domains
`tag2name`	Prints the pathname of a file given the tag number
`verify`	Checks for and repairs file system inconsistencies

Table B–5: AdvFS Quota Commands

Command	Description
`chfsets`	Changes file and block quotas; turns object safety, fragging, and data management API on and off
`edquota`	Edits user and group quotas
`quot`	Summarizes file and block information
`quota`	Displays disk usage and limits by user or group
`quotacheck`	Checks file system quota consistency
`quotaoff`	Turns quotas off

Table B–5: AdvFS Quota Commands (cont.)

Command	Description
`quotaon`	Turns quotas on
`repquota`	Summarizes quotas for a file system

Table B–6: AdvFS Disk Structure Commands

Command	Description
`nvbmtpg`	Displays a formatted page of the bitfile metadata table (BMT)
`nvfragpg`	Displays file fragment information
`nvlogpg`	Displays a formatted page of the log
`nvtagpg`	Displays a formatted page of the tag directory
`savemeta`	Captures a domain's metadata
`shblk`	Displays unformatted disk blocks
`shfragbf`	Displays file fragment information
`vfilepg`	Displays a page of an AdvFS file
`vsbmpg`	Displays a page of the storage bitmap

B.2 AdvFS Utilities Commands

The following table lists and describes AdvFS Utilities commands. These commands require the optional AdvFS Utilities product license. If you installed the AdvFS Utilities reference page subset, you can access reference pages for each of these commands by entering the `man` command.

Table B–7: AdvFS Utilities Commands

Command	Description
`addvol`	Adds a volume to an existing domain
`advfsd`	Starts the AdvFS GUI agent (daemon)
`balance`	Balances the percentage of used space between volumes
`clonefset`	Creates a read-only copy of a fileset
`dtadvfs`	Starts the AdvFS GUI
`lsmsa`	Starts Logical Storage Manager Storage Administrator (LSMSA)
`migrate`	Moves a file to another volume in the domain

Table B–7: AdvFS Utilities Commands (cont.)

Command	Description
`mktrashcan`	Attaches directories to a trashcan directory, which stores deleted files
`rmtrashcan`	Detaches a specified directory from a trashcan directory
`rmvol`	Removes a volume from an existing domain
`shtrashcan`	Shows the trashcan directory, if any, that is attached to a specified directory
`stripe`	Interleaves storage allocation of a file across two or more volumes within a domain

B.3 Comparison of AdvFS and UFS Commands

The following table lists the AdvFS commands and equivalent or similar commands available in UFS file systems. Note that there is no one-to-one correspondence between all AdvFS and all UFS commands.

Table B–8: Comparison of AdvFS and UFS Commands

AdvFS Command	UFS Command
`addvol`	None; UFS does not use volumes.
`advfsd`	None; no UFS GUI.
`advfsstat`	Use `iostat` or system monitoring tools such as `sys_check` and *Collect for Tru64 UNIX*. Use event monitoring tools to watch for file system events.
`advscan`	None; UFS does not use volumes.
`balance`	None; UFS does not use volumes.
`chfile`	None; the only attributes that can be changed are ownership and protection.
`chfsets`	None; the only attributes that can be changed are ownership and protection.
`chvol`	None; UFS does not use volumes.
`clonefset`	None.
`defragment`	None.
`dtadvfs, lsmsa`	None; no UFS GUI.
`migrate`	Use `cp` and `mv`.
`mkfdmn, mkfset`	Use `newfs`.

Table B–8: Comparison of AdvFS and UFS Commands (cont.)

AdvFS Command	UFS Command
mktrashcan, rmtrashcan, shtrashcan	Use CDE windowing environment, which provides a deferred deletion.
mountlist	Use mount.
ncheck	Same as AdvFS.
fixfdmn, nvbmptg, nvfragpg, nvlogpg, nvtagpg, savemeta, shblk, shfragbf, vfilepg, vsbmpg	None; no analogous UFS commands.
quota, edquota, quot, quotacheck, quotaoff, quotaon, repquota	Same as AdvFS.
renamefset	None.
rmfdmn, rmfset	Unmount the file system using mount, remove the entry from the /etc/fstab file.
rmvol	None; UFS does not use volumes.
vdump, rvdump, vrestore, rvrestore	Use dump, rdump, restore, and rrestore. Can use AdvFS commands.
salvage	None.
showfdmn	None.
showfile	Use file and ls -l.
stripe	None.
switchlog	None.
tag2name	None.
vdf	Use df and du.
verify	Use fsck and bcheckrc.

C
Converting File Systems

This appendix contains procedures you can use to convert a /usr file system, the root file system, and a data file system to AdvFS. It also includes instructions for converting your entire system from AdvFS to UFS.

The methods provided here are guidelines; that is, they are suggestions that illustrate the process of conversion. Specific file names, tape drives, and disk partitions depend on your system.

The vdump and vrestore file formats are compatible for Version 4 and Version 5.0 of the operating system software. If you upgrade your system, recreate the domains, and restore the data from backup (see Chapter 3), your AdvFS filesets and domains have the updated structure (see Section 1.6.3).

C.1 Converting a /usr File System to AdvFS

During the initial installation of AdvFS, you can use AdvFS for the /usr file system. Converting the /usr (UFS) file system to AdvFS reduces the amount of time your system is down after a system failure. If you have not installed /usr on AdvFS, you can do so with a backup tape, an intermediate file, or from a second disk.

C.1.1 Using a Backup Tape

You can convert the /usr (UFS) file system to an equivalent AdvFS file system by backing up the existing file system to tape and restoring it to an AdvFS environment.

The following are required:

- Root user privilege
- Backup device and media
- Five percent more disk space for the converted file system
- AdvFS installed on your system

Assumed system configurations are as follows:

- Existing UFS configuration:

  File system /usr

  Disk partition /dev/disk/dsk3g

- New AdvFS configuration:

  File system /usr

  Disk partition /dev/disk/dsk3g

  Domain usr_domain

  Fileset usr

Use the following procedure as a guide for converting the file system:

1. Log in as root on the system containing the /usr file system.
2. Use the AdvFS vdump command to back up the /usr file system to /dev/tape/tape0:

   ```
   # mt rewind
   # cd /usr
   # vdump -0 .
   ```

3. Edit the /etc/fstab file:

   a. Search for the entry that mounts /usr as a UFS file system, such as:

   ```
   /dev/disk/dsk3g /usr ufs rw 1 2
   ```

   b. Replace it with one that mounts /usr as an AdvFS file system:

   ```
   usr_domain#usr /usr advfs rw 1 0
   ```

4. Shut down the system:

   ```
   # shutdown -h now
   ```

5. Reboot the system in single-user mode. See *System Administration* for instructions on invoking single-user mode.
6. In single-user mode, mount the root file system as rw. Create the usr_domain domain, and create the usr fileset. Use the mkfdmn -F command to force the partition label to change from UFS to AdvFS and to avoid a warning message.

   ```
   # mount -u /
   # mkfdmn -F /dev/disk/dsk3g usr_domain
   # mkfset usr_domain usr
   ```

7. Mount the usr fileset on the /usr directory:

   ```
   # mount -t advfs usr_domain#usr /usr
   ```

8. Restore the /usr file system from tape to the usr fileset:

   ```
   # vrestore -x -D /usr
   ```

9. Boot the system to multiuser mode. When the system prompt returns, the converted /usr file system is ready to use.

C.1.2 Using an Intermediate File

You can convert the /usr (UFS) file system to the equivalent AdvFS file system by backing up the existing file system to a file and restoring it to an AdvFS environment.

The following are required:

- Root user privilege
- Disk space (on a different file system) for an intermediate file. (The file system containing the intermediate file can be on the same disk or a different disk.)
- Five percent more disk space for the converted file system
- AdvFS installed on your system

Assumed system configurations are as follows:

- Existing UFS configuration:

  File system /usr

  Disk partition /dev/disk/dsk3g

  Intermediate file /tmp/usr_bck

- New AdvFS configuration:

  File system /usr

  Disk partition /dev/disk/dsk3g

  Domain usr_domain

  Fileset usr

Use the following procedure as a guide for converting the /usr file system:

1. Log in as root on the system containing the /usr file system.
2. Use the AdvFS vdump command to back up the /usr file system to /tmp/usr_bck, the intermediate file:

   ```
   # cd /usr
   # vdump -0f /tmp/usr_bck /usr
   ```

3. Edit the /etc/fstab file:

   a. Search for the entry that mounts /usr as a UFS file system:

      ```
      /dev/disk/dsk3g /usr ufs rw 1 2
      ```

   b. Replace it with one that mounts /usr as an AdvFS file system:

      ```
      usr_domain#usr /usr advfs rw 1 0
      ```

4. Shut down the system:

```
# shutdown -h now
```

5. Reboot the system in single-user mode. See *System Administration* for instructions on invoking single-user mode.
6. In single-user mode, mount the root file system as `rw`, create the `usr_domain` domain, and create the `usr` fileset. Use the `mkfdmn -F` command to force the partition label to change from UFS to AdvFS and to avoid a warning message.

   ```
   # mount -u /
   # mkfdmn -F /dev/disk/dsk3g usr_domain
   # mkfset usr_domain usr
   ```

7. Mount the `usr` fileset on the `/usr` directory:

   ```
   # mount -t advfs usr_domain#usr /usr
   ```

8. Restore the `/usr` file system from the intermediate file to the `usr` fileset:

   ```
   # vrestore -xf /tmp/usr_bck -D /usr
   ```

9. Boot the system to multiuser mode. When the system prompt returns, the converted `/usr` file system is ready to use.

C.1.3 Using a Second Disk

You can convert the `/usr` (UFS) file system on one disk to the equivalent `/usr` (AdvFS) file system on a different target disk.

The following are required:

- Root user privilege
- A second disk labeled `unused` with 5% more disk space for the converted file system
- AdvFS installed on your system

Assumed system configurations are as follows:

- Existing UFS configuration:

  File system `/usr`

  Disk partition `/dev/disk/dsk3g`

- New AdvFS configuration:

  File system `/usr`

  Disk partition `/dev/disk/dsk2c`

  Mount directory `/usr.advfs`

  Domain `usr_domain`

  Fileset `usr`

Use the following procedure as a guide for converting the /usr file system:

1. Log in as root on the system containing the /usr file system.
2. Create a domain and fileset:

   ```
   # mkfdmn /dev/disk/dsk2c usr_domain
   # mkfset usr_domain usr
   ```

3. Create a mount-point directory and mount the new fileset on the directory:

   ```
   # mkdir /usr.advfs
   # mount -t advfs usr_domain#usr /usr.advfs
   ```

4. Change to the /usr directory:

   ```
   # cd /usr
   ```

5. Be certain there is no activity on the system. (You can do this by bringing the system to single-user mode.) Copy the contents of the UFS file system to the AdvFS file system:

   ```
   # vdump -0f - -D . | vrestore -xf - -D /usr.advfs
   ```

6. Edit the /etc/fstab file:

   a. Search for the entry that mounts /usr as a UFS file system, such as:

      ```
      /dev/disk/dsk3g /usr ufs rw 1 2
      ```

   b. Replace it with one that mounts /usr as an AdvFS file system:

      ```
      usr_domain#usr /usr advfs rw 1 0
      ```

7. Remove the temporary directory:

   ```
   # umount /usr.advfs
   # rmdir /usr.advfs
   ```

8. Shut down and reboot the system. When the system prompt returns, the converted /usr file system is ready to use.

C.2 Converting the root File System to AdvFS

By converting the root file system to AdvFS, you can boot your system from an AdvFS domain and use AdvFS as the root (/) file system. The AdvFS root domain must reside on a single disk. During initial installation you can install root on the AdvFS file system. If you do not, you can use the following method.

Note

Before you begin the conversion, check the size of the existing UFS root partition. The target AdvFS root domain can contain only one volume and must be large enough to accommodate the converted root file system.

You must have root user privilege, a second bootable disk (use partition a or c) and the AdvFS file system installed on your system.

Assumed system configurations are as follows:

- Existing UFS configuration:

  File system root

  Mount directory /newroot

  Disk partition /dev/disk/dsk1a

- New AdvFS configuration:

  File system root

  Mount directory /newroot

  Disk partition /dev/disk/dsk2a

  Domain root_domain

  Fileset root

Use the following procedure as a guide for converting the root file system. This example assumes a device called <disk>.

1. Log in as root on the system containing the root file system.
2. Create a domain and fileset:

   ```
   # mkfdmn -r /dev/disk/dsk2a root_domain
   # mkfset root_domain root
   ```

3. Create a mount-point directory and mount the new fileset on the directory:

   ```
   # mkdir /newroot
   # mount -t advfs root_domain#root /newroot
   ```

4. Be certain there is no activity on the system. (You can do this by bringing the system to single-user mode.) Restore the UFS root file system to the root fileset:

   ```
   # vdump 0f - / | (cd /newroot; vrestore -xf -)
   ```

5. Make the disk with the root domain a bootable disk:

   ```
   # disklabel -r /dev/rdisk/dsk2a > /tmp/dsk2label
   # disklabel -t advfs -r -R /dev/rdisk/dsk2a \
   ```

```
/tmp/dsk2label <disk>
```

6. Edit the /etc/fstab file on the AdvFS root fileset to indicate the new root entry.

   a. Search /newroot/etc/fstab for the entry that mounts root as a UFS file system, such as:

   ```
   /dev/disk/dsk1a / ufs rw 1 1
   ```

   b. Replace it with one that mounts root as an AdvFS file system:

   ```
   root_domain#root / advfs rw 1 0
   ```

7. After editing is complete, shut down the system:

   ```
   # shutdown -h now
   ```

8. Reset the boot default device, BOOTDEF_DEV, to point to the disk with the new root domain. This procedure is hardware-specific. Refer to your hardware manual for instructions.
9. Reboot the system to enable the AdvFS root file system.

The converted root file system is ready to use.

Because the AdvFS root domain is limited to one disk, you cannot use the addvol command to extend the root domain.

C.3 Converting a Data File System to AdvFS

By converting your data file systems to AdvFS, you can eliminate lengthy reboots. Moreover, you can easily modify your file system configurations to meet changing system requirements.

To convert data file systems from UFS to AdvFS, you need a backup tape, an intermediate file, a second disk, or a second system.

C.3.1 Using a Backup Tape

You can convert a data (UFS) file system to the equivalent data (AdvFS) file system by backing up the existing file system to tape by using the vdump command and restoring it with the vrestore command to an AdvFS environment.

The following are required:

- Root user privilege
- Backup device and media
- Five percent more disk space for the converted file system
- AdvFS installed on your system

Assumed system configurations are as follows:

- Existing UFS configuration:

  File system /staff2

  Mount directory /staff2

  Disk partition /dev/disk/dsk2c

- New AdvFS configuration:

  File system /staff2

  Disk partition /dev/disk/dsk2c

  Domain staff_domain

  Fileset staff2

Use the following procedure as a guide for converting the /staff2 file system:

1. Log in as root on the system containing the /staff2 file system.
2. Use the AdvFS vdump command to back up the /staff2 file system to /dev/tape/tape0, the default tape drive:

   ```
   # mt rewind
   # mount /staff2
   # vdump -0f /dev/tape/tape0 /staff2
   # umount /staff2
   ```

3. Create the staff_domain domain and the staff2 fileset. Use the mkfdmn -F command to force the partition label to change from UFS to AdvFS and to avoid a warning message.

   ```
   # mkfdmn -F /dev/disk/dsk2c staff_domain
   # mkfset staff_domain staff2
   ```

4. Mount the new fileset on the directory:

   ```
   # mount -t advfs staff_domain#staff2 /staff2
   ```

5. Restore the /staff2 file system from tape to the staff2 fileset:

   ```
   # vrestore -xvf /dev/tape/tape0 -D /staff2
   ```

6. Edit the /etc/fstab file:

   a. Search for the entry that mounted /staff2 as a UFS file system:

   ```
   /dev/disk/dsk2c /staff2 ufs rw 1 2
   ```

   b. Replace it with one that mounts /staff2 as an AdvFS file system:

   ```
   staff_domain#staff2 /staff2 advfs rw 1 0
   ```

The converted /staff2 file system is ready to use.

C.3.2 Using an Intermediate File

You can convert a data (UFS) file system to the equivalent data (AdvFS) file system by backing up the existing file system to a file and restoring it to an AdvFS environment.

The following are required:

- Root user privilege.
- Disk space (on a different file system) for an intermediate file. (The file system containing the intermediate file can be on the same disk or a different disk.)
- Five percent more disk space for the converted file system.
- AdvFS installed on your system

Assumed system configurations are as follows:

- Existing UFS configuration:

  File system `/staff2`

  Disk partition `/dev/disk/dsk3g`

  Intermediate file `/tmp/staff_bck`
- New AdvFS configuration:

  File system `/staff2`

  Disk partition `/dev/disk/dsk3g`

  Domain `staff_domain`

  Fileset `staff2`

Use the following procedure as a guide for converting the `/staff2` file system:

1. Log in as root on the system containing the `/usr` file system.
2. Use the AdvFS `vdump` command to back up the `/staff2` file system to `/tmp/staff_bck`, the intermediate file:

   ```
   # vdump -0f /tmp/staff_bck /staff2
   ```
3. Create the `staff_domain` domain and the `staff2` fileset. Use the `mkfdmn -F` command to force the partition label to change from UFS to AdvFS and to avoid a warning message.

   ```
   # mkfdmn -F /dev/disk/dsk3g staff_domain
   # mkfset staff2
   ```
4. Edit the `/etc/fstab` file:

   a. Search for the entry that mounts `/staff2` as a UFS file system:

```
/dev/disk/dsk3g /staff2 ufs rw 1 2
```

b. Replace it with one that mounts `/staff2` as an AdvFS file system:

```
/staff_domain#staff2 /staff2 advfs rw 1 0
```

5. Mount the `staff2` fileset on the `/staff2` directory by entering the following command:

```
# mount -t advfs staff_domain#staff2 /staff2
```

6. Restore the `/staff2` file system from the intermediate file to the `staff2` fileset:

```
# vrestore -xf /tmp/staff_bck -D /staff2
```

The converted `/staff2` file system is ready to use.

C.3.3 Using a Second Disk

You can convert a data (UFS) file system on one disk to the equivalent data (AdvFS) file system on a different target disk.

The following are required:

- Root user privilege
- A second disk with 5% more disk space for the converted file system
- AdvFS installed on your system

Assumed system configurations are as follows:

- Existing UFS configuration:

  File system `/staff2`

  Disk partition `/dev/disk/dsk3g`

- New AdvFS configuration:

  File system `/staff2`

  Disk partition `/dev/disk/dsk2c`

  Mount directory `/staff2`

  Domain `staff_domain`

  Fileset `staff2`

Use the following procedure as a guide for converting the `/staff2` file system:

1. Log in as root on the system containing the `/staff2` file system.
2. Create the `staff_domain` domain and `staff2` fileset:

```
# mkfdmn /dev/disk/dsk2c staff_domain
# mkfset staff_domain staff2
```

3. Create a mount-point directory and mount the new fileset on the directory:

   ```
   # mkdir /new_staff2
   # mount -t advfs staff_domain#staff2 /new_staff2
   ```

4. Be certain there is no activity on the system. Copy the contents of the UFS file system to the AdvFS file system:

   ```
   # vdump -0f - -D /staff2 | vrestore -xf - -D /new_staff2
   ```

5. Edit the `/etc/fstab` file:

   a. Search for the entry that mounts `/staff2` as a UFS file system, such as:

      ```
      /dev/disk/dsk3g /staff2 ufs rw 1 2
      ```

   b. Replace it with one that mounts `/staff2` as an AdvFS file system:

      ```
      staff_domain#staff2 /staff2 advfs rw 1 0
      ```

6. Unmount `/new_staff2`:

   ```
   # umount /new_staff2
   ```

7. Unmount `/staff2`:

   ```
   # umount /staff2
   ```

8. Remove the old directory:

   ```
   # rmdir /staff2
   ```

9. Mount the new fileset:

   ```
   # mount /staff2
   ```

C.3.4 Using a Second System

You can transfer an existing data file system to a new system, then you can convert the file system to AdvFS.

The following are required:

- Two systems and a common facility for transferring the files such as the `tar` utility (see `tar`(1))
- Root user privilege on the target system
- Five percent more disk space for the converted file system
- AdvFS installed on the target system

Assumed system configurations are as follows:

- Existing UFS configuration:

  File system /staff4

- New AdvFS configuration:

  File system /staff4

  Disk partition /dev/disk/dsk2c

  Mount directory /staff4

  Domain staff_domain

  Fileset staff4

Use the following procedure as a guide for converting the staff4 file system:

1. Log in to the system containing the /staff4 file system and back up the file system to tape:

   ```
   # tar c /staff4
   ```

2. Log in as root user on the target system.
3. Create the staff_domain domain and the staff4 fileset:

   ```
   # mkfdmn /dev/disk/dsk2c staff_domain
   # mkfset staff_domain staff4
   ```

4. Create a mount-point directory and mount the new fileset on the directory:

   ```
   # mkdir /staff4
   # mount -t advfs staff_domain#staff4 /staff4
   ```

5. Restore the /staff4 file system from the default tape drive, /dev/tape/tape0:

   ```
   # mt rewind
   # tar x /staff4
   ```

6. Edit the /etc/fstab file to add an entry that mounts /staff4 as a UFS file system:

   ```
   staff_domain#staff4 /staff4 advfs rw 1 0
   ```

The staff_domain domain now includes the staff4 fileset, which is ready to use.

C.4 Converting from AdvFS to UFS

Converting your entire system from AdvFS to UFS is a multistep process. You first convert the AdvFS root file system to UFS. Then, you convert each AdvFS fileset to a UFS file system.

C.4.1 Converting the root File System to UFS

To convert the root file system, you must mount a UFS disk while your AdvFS root fileset is mounted.

The following are required:

- Root user privilege
- A second bootable disk (You must use partition `a`.)

Assumed system configurations are as follows:

- Existing AdvFS configuration:

  File system `root`

  Disk partition `/dev/disk/dsk1a`

  Domain `root_domain`

  Fileset `root`

- New UFS configuration:

  File system `root`

  Disk partition `/dev/disk/dsk2a`

Use the following procedure as a guideline for converting your file system:

1. Log in as root user.
2. Create a UFS file system (on a device here called <disk>):

   ```
   # newfs /dev/disk/dsk2a <disk>
   ```

3. Create a mount-point directory and mount the UFS file system:

   ```
   # mkdir /newroot
   # mount -t ufs /dev/disk/dsk2a /newroot
   ```

4. Restore the AdvFS root file system to the `/dev/disk/dsk2a` UFS file system:

   ```
   # vdump -0f - / | (cd /newroot; vrestore -xf -)
   ```

5. Make the disk containing the UFS file system a bootable disk:

   ```
   # disklabel -r /dev/rdisk/dsk2a > /tmp/dsk21abel
   # disklabel -t ufs -r -R /dev/rdisk/dsk2a \
   /tmp/dsk21abel <disk>
   ```

6. Edit the `/etc/fstab` file on the UFS file system to refer to the new root entry.

   a. Search `/newroot/etc/fstab` for the entry previously mounted as root for the AdvFS file system:

      ```
      root_domain#root / advfs rw 1 0
      ```

b. Replace the entry with one that mounts root as a UFS file system:

```
/dev/disk/dsk2a / ufs rw 1 1
```

7. Shut down the system by entering the following command:

```
# shutdown -h now
```

8. Reset the boot default device, `BOOTDEF_DEV`, to the new root disk. (Refer to your hardware manual for specific information.)
9. Reboot the system to enable the UFS root file system.

C.4.2 Converting a Fileset to UFS

After the root file system is converted to UFS, you can convert your filesets.

The following are required:

- Root user privilege
- A tape or disk for back up

Assumed system configurations are as follows:

- Existing AdvFS configuration:

  Mount directory `/staff2`

  Domain `staff_domain`

  Fileset `staff2`
- New UFS configuration:

  Mount directory `/staff2`

  Disk partition `/dev/disk/dsk2c`

The following example assumes your AdvFS domain contains one volume and only one fileset. If the AdvFS domain contains multiple filesets, then you must create a separate UFS file system for each fileset.

Caution

Be sure to perform a full backup on all AdvFS filesets before you start the conversion.

1. Make a backup of the AdvFS fileset:

```
# vdump -0f /dev/tape/tape0 /staff2
```

2. Unmount the fileset:

```
# umount /staff2
```

3. Delete the fileset:

```
# rmfset staff_domain staff2
```

4. Remove the domain:

   ```
   # rmfdmn staff_domain
   ```

5. Create the UFS file system for the specified disk type (on a device here called <disk>). (If the partition you are creating is currently labeled AdvFS, you get a warning message. Respond **yes** to override the AdvFS designation.)

   ```
   # newfs /dev/disk/dsk2c <disk>
   ```

6. Edit your /etc/fstab file:

   a. Search for the entry that mounts /staff2 as an AdvFS fileset, such as:

      ```
      staff_domain#staff2 /staff2 advfs rw 1 0
      ```

   b. Replace it with one that mounts /staff2 as a UFS file system:

      ```
      /dev/disk/dsk2c /staff2 ufs rw 1 2
      ```

7. Mount the UFS file system:

   ```
   # mount -t ufs /dev/disk/dsk2c /staff2
   ```

8. Use the vrestore command to load the files from the backup into the UFS file system:

   ```
   # vrestore -xvf /dev/tape/tape0 -D /staff2
   ```

If your domain contains multiple volumes, you must verify that the disk space allocated to a fileset does not exceed the limit of the UFS file system disk partition. You might need to create multiple UFS file systems to hold the filesets in the domain.

D

Graphical User Interfaces

There are two interfaces that provide graphical representation of the AdvFS file system: the Logical Storage Manager Storage Administrator (LSMSA) and the AdvFS Graphical User Interface (AdvFS GUI). LSMSA allows you to manage LSM objects and perform LSM as well as the AdvFS file system administrative functions. In contrast, the AdvFS Graphical User Interface GUI provides a visual representation of the AdvFS file system. Both require utilities licenses.

LSMSA is a Java-based application that operates on local or remote (client) systems. LSMSA operations are described in detail in *Logical Storage Manager*.

AdvFS GUI operations are described below. The AdvFS GUI is designed to run under the Common Desktop Environment (CDE). You can use the GUI locally or as a remote system manager.

To access the AdvFS GUI, you must register the AdvFS Utilities license and you must have root user privilege. For information about license activation contact your software support organization.

D.1 Installing the AdvFS GUI

To load the AdvFS GUI, choose the AdvFS Utilities subset when you update or install the operating system.

Table D–1 contains the subset titles, names (where xxx is the subset identification number), and descriptions of the subsets that make up the AdvFS GUI. The disk space requirements for loading and running AdvFS Utilities software subsets are shown in Table D–2.

Table D–1: AdvFS GUI Subsets

Subset	Name	Description
AdvFS Utilities	(OSFADVFSxxx)	Contains a set of advanced utilities licensed for managing AdvFS
AdvFS Graphical User Interface (dtadvfs)	(OSFXADVFSxxx)	Contains the AdvFS Graphical User Interface and online help files

Table D–1: AdvFS GUI Subsets (cont.)

Subset	Name	Description
AdvFS Agent (advfsd)	(OSFADVFSDAEMONxxx)	Contains the AdvFS agent, which runs in the traditional style of the UNIX daemon
AdvFS Japanese Graphical User Interface	(OISJPXADVFSxxx)	Localization files that must also be loaded when the Japanese version of the GUI is run

Table D–2: Disk Space Requirements for the AdvFS GUI

Utility	/(root)	/var	/usr
Utilities Subset	0	0	300
GUI (`dtadvfs`)	30	20	7000
Agent (`advfsd`)	0	50	800
Japanese localization files	0	1	661

D.2 Components of the GUI

There are two parts to the GUI: `advfsd`, the agent, and `dtadvfs`, the actual graphical interface. The agent must be running for the GUI to operate.

D.2.1 GUI Agent (advfsd)

The AdvFS GUI **agent**, **advfsd**, issues commands and obtains system information for the GUI. If the agent is not running, the GUI cannot operate. The agent is automatically started at boot time and when the OSFADVFSDAEMON subset is installed. It runs unseen in the background.

Under normal conditions, `advfsd` does not need to be run manually. If you want to start or stop the agent at any time, do so from the command line (see `advfsd`(8)). If you are not running Performance Manager, you can stop the agent and restart it when you run the GUI.

To stop the agent:

```
# /sbin/init.d/advfsd stop
```

To restart the agent:

```
# /sbin/init.d/advfsd start
```

Only one agent can be running on a system at a given time. If you attempt to start a second copy of `advfsd`, it fails.

The agent allows Simple Network Management Protocol (SNMP) clients such as NetView to request AdvFS information. This is not a two-way path: SNMP clients cannot issue system configuration commands to `advfsd`.

The agent periodically asks for information from the system on which it is running. It then updates the file systems and the storage device information passed to the GUI display. It also evaluates any free space alert conditions. The time interval for when the agent scans the system disks is the **agent state monitor interval**. To change the agent state monitor interval, see Section D.3.1. The interval is saved between restarts of the agent.

D.2.2 AdvFS GUI (dtadvfs)

Only the root user can start the AdvFS GUI, **dtadvfs**. It can be started in a terminal window or from a CDE icon. The file system operations that the GUI can perform are described in Section D.4.

To start the GUI from the command line, enter:

```
# /usr/bin/X11/dtadvfs &
```

To start the GUI using CDE icons:

1. Select the Application Manager on the CDE toolbar.
2. Double-click on the System_Admin icon in the Application Manager window.
3. Double-click on the Storage_Management icon in the System_Admin window.
4. Double-click on the Advanced File System icon in the Storage_Management window.

When the GUI is running, window displays are periodically updated to reflect the changing characteristics of the file system. The **GUI refresh interval**, the time interval between updates, is 15 minutes by default. To change the GUI refresh interval when the GUI is running, see Section D.3.1. The GUI refresh interval is not saved between restarts of the GUI.

D.2.2.1 GUI Security

Two optional security files are associated with each system that the GUI manages. The password file restricts the use of the GUI managing a particular system to administrators who know the password. The allowable hosts file allows only GUIs running on systems listed in the file to remotely manage it. These files are configured through an editor of your choice. They cannot be accessed directly from the GUI.

The root user creates the password file, /var/advfs/daemon/socket/gui.passwd, on the system for which the password is to be used. There is one password file containing one password per system. (It is not a good idea to use the root password.) The file contains the password in plain text. Only users who enter the correct password can run a GUI connected to the system. A user wanting to use the GUI to manage the system from a remote location must know this password in order to connect to the system.

The allowable hosts file, /var/advfs/daemon/socket/hosts.allow, is also created by the root user. It contains a plain text list of all systems (hosts) on which a GUI can operate to manage the system. The hosts.allow file automatically includes the GUI running locally on the system; that is, it includes itself. If you want to allow others to remotely manage your system, you must include them in your hosts.allow file to authorize the agent to send your file system information to their system. If you have protected your system with a password in the system's gui.passwd file, remote users also have to know that password.

See advfsd(8) and dtadvfs(8) for more information.

D.2.2.2 Ignoring Disks

Create the /var/advfs/daemon/disks.ignore file to specify a list of disks that the agent does not examine. The disks.ignore file contains a plain-text list of disk drives, one per line. This file is useful because performance might be reduced when there are off-line HSZ devices or spun-down disk drives.

Since the agent processes the disks.ignore file every time the disks are checked, disks that fail can be added to the file and disks listed in the file that become available can be removed. It is not necessary to stop the agent. See advfsd(8) for more information.

You cannot ignore an LSM volume by including the LSM volume name in the disks.ignore file; you must list the disks from which the LSM volume is built. To ignore a complete disk group, you must list all disks in it. Because all partitions on the listed disks are ignored, unexpected results might occur if a disk has partitions belonging to more than one disk group.

D.2.2.3 Log Files

Entries are generated in the agent log file, /var/advfs/daemon/logs/advfsd, and the GUI log file, /var/advfs/gui/advfs_gui.log, as processing proceeds. It is a good idea to periodically check these files. Both the agent and the GUI log files are periodically renamed to prevent unrestricted consumption of disk space.

D.2.2.4 Help

The help system contains background information about the AdvFS file system as well as specific information about the GUI. Each operation is described. Detailed explanations of the dialog boxes that are used to perform file system activities are provided.

D.2.2.5 Monitoring Remote File Systems

From your system you can remotely monitor file systems that have your system listed in their `hosts.allow` file. You can attach to these systems through the GUI:

1. Choose Host... from the AdvFS menu of the main window.
2. The Select a Host dialog box appears, in which you can select another host system or choose to modify the host list to add or delete host systems. This list is created by you and saved on your system to identify the hosts you connect to regularly.
3. Select the host and click Connect.

If you try to connect to a host but are unsuccessful, you are no longer connected to any host so you must repeat the connection process for another system.

Note that the host list is for convenience only. It does not provide additional security. You can always type a host name in the Selection block of the Select a Host dialog box.

D.2.2.6 Managing LSM Volumes

If you are using LSM volumes, it is important to know how the GUI treats these volumes:

- An LSM volume is listed as available if it is labeled `unused`.
- Do not use LSM to change the size of an LSM volume that is in use by an AdvFS domain. AdvFS is unable to accommodate this change.
- If you added the LSM volume `/dev/vol/volname` to the domain from the command line, the AdvFS GUI does not recognize that it is the same as `/dev/vol/rootdg/volname`. If the volume has mounted filesets, it appears as unknown in the devices and volumes window; if it has no mounted filesets, it appears as unavailable. If you try to add it to a domain, you get an error message saying it is in use.
- If you encapsulate a volume in use by AdvFS into LSM while the GUI is running, the volume name is not updated in the GUI window. You must exit and restart the GUI to display the correct name.

- You cannot ignore an LSM volume by including the LSM volume name in the `disks.ignore` file; you must list the disks from which the LSM volume is built.

D.3 Displaying File System Information

The **objects** (components) that make up the file system (volumes, domains, filesets, and fileset clones) are viewed in the GUI windows in an **object tree** that shows their hierarchical relationship to each other. If an object contains other objects, a folder icon appears to its left. Clicking on the folder icon expands or collapses the object tree; that is, displays or hides the objects below it in the tree. For example, clicking on a domain folder icon displays or hides the filesets that belong to that domain.

You can change your window views and look at critical information about the objects in a number of ways. You can view:

- Main window information

  The first window that appears, the main window, provides a comprehensive view of the file system. It is the starting point for all GUI-managed tasks and the primary screen for monitoring file system status. Use the items on the View menu to display the status of domains and filesets; domains and volumes; and domains, filesets, and volumes. You can also use the View menu to change from one of these views to viewing fileset quotas and back.

- Devices and volumes

  Choose Devices and Volumes from the Configuration menu for a different detailed view. The devices and volumes window displays all the physical disks, disk groups, and logical volumes on the system and can be used to determine volume type or partition availability.

- Object characteristics

  Choose Show... from the Configuration menu to display more complete information about the object that you have highlighted in the main window display.

  You can usually select Show... as an alternate way of performing the file system management tasks described in the following sections. For example, highlight the domain and choose Show... to see an object dialog box that includes the creation date and number of log pages for that domain.

Note that double-clicking a domain, fileset, clone, or volume in the object tree is the same as choosing Show... for that object. Double-clicking does not work for hosts, devices, or partitions or for the volume icon in the devices and volumes window. Single-clicking an object in an object tree only highlights it. No new information is presented.

Clicking the right mouse button on an object brings up a list of commands appropriate for that object.

If you attempt to access a menu item and find it unavailable (grayed out), check that the appropriate object is highlighted. For example, you need to select a fileset before you can create a clone for it.

D.3.1 Choosing Units for Displaying Information

You can customize the display of your file system information by choosing Options from the AdvFS menu. All options except the agent state monitor interval are reset each time the GUI is restarted.

- Units

  Choose to display disk space usage in 512–kilobyte blocks, kilobytes, megabytes, gigabytes, or terabytes.

- Update Interval

  Choose the GUI refresh rate.

  Choose the Agent state monitor interval.

- Free Space Alert

  Choose between space used > and free space <.

  Choose to display space available as a percent or in the units you have chosen for the display.

- Fileset quotas

  Choose to calculate quotas by percent of space or by the amount of space used.

D.3.2 Main Window

When the main window first displays, disk-space usage information is shown for the host and for domains. Options on the View menu let you display domains, filesets, and volumes in different combinations.

From the main window you can also view fileset quota information by choosing Fileset Quotas from the View menu. You can return to viewing domain, fileset, and volume information by selecting the items you want to view from the same View menu.

The main window has menu selections that you can use to accomplish file system management tasks for domains, filesets, clones, and volumes. Menu commands call up dialog boxes to help you accomplish these tasks.

D.3.3 Devices and Volumes Window

The devices and volumes window displays disk groups and logical volumes. It identifies the size and type of disks and what each partition is used for. This view is particularly useful if you want to enlarge your domain and need to locate a volume to use. From the View menu of the devices and volumes window, you can sort your objects, choose to display only available volumes, or choose to display overlapped volumes.

D.4 Performing File System Operations

You can perform file system management tasks on domains, filesets, AdvFS fileset clones, and volumes directly from the GUI. In general, you must select the appropriate object before you can proceed. In the following sections, file domain and domain are synonymous.

D.4.1 Operations on a Domain

File system management tasks for domains are initiated from the main window by selecting Show or File Domain on the Configuration menu. In addition, from the devices and volumes window you can create a new domain and add volumes to an existing domain.

D.4.1.1 Creating a Domain

To create a domain, you must assign a name and select an initial volume. You can set a free space alert.

A domain is not completely defined until you have created at least one fileset. A domain is not active unless there is a mounted fileset. To create a domain using the main window:

1. Choose File Domain from the Configuration menu.
2. Choose New from the File Domain menu.
3. In the New File Domain dialog box, enter the domain name and highlight an available volume in the object tree. You might need to expand the device list by clicking on the folder to the left of the device name. You can also set the free space alert.

If you are not certain what storage device to use for the new domain, work from the devices and volumes window (on the Configuration menu) to get a complete view of the storage on your system:

1. Choose Devices and Volumes from the Configuration menu of the main window.

2. In the devices and volumes window, choose Show available volumes only from the View menu.
3. Choose a volume that is labeled as available.
4. Choose New File Domain from the Configuration menu.
5. In the New File Domain dialog box, create the domain and set the free space alert. If you are monitoring a file system prior to Version 4.0E, see the main window instructions for directions on using the Advanced button.

Creating a domain automatically generates an entry in the `/etc/fdmns` directory.

D.4.1.2 Setting a Free Space Alert for a Domain

Set a free space alert to inform you when the free space threshold in the domain is reached or passed. When you set the alert, you can choose to automatically run a script when the alert threshold is crossed. See `/usr/advfs/daemon/scripts` for examples of scripts used by the agent to execute commands.

To set the free space alert from the main window:

1. In the object tree, highlight the domain.
2. Choose File Domain from the Configuration menu.
3. Choose Set free space alert from the File Domain menu.
4. In the File Domain dialog box, set the free space alert.

You must reset the free space alert when the size of a domain changes. Free space alert values are evaluated using actual size (for example, kilobytes), not percentage of space.

D.4.1.3 Enlarging a Domain

Add volumes to transform a single-volume domain (except the root domain, which can only have one volume) into a multivolume domain or to enlarge a multivolume domain (see Section 1.6.6).

To add a volume to a domain:

1. In the object tree, highlight the domain.
2. Choose File Domain from the Configuration menu.
3. Choose Add Volume from the File Domain menu.

4. In the Add Volume dialog box, highlight an available volume to add. You might need to expand the view of the volumes by clicking the folder icon to the left of the device name.

If you do not know which volume to add to your domain, work from the devices and volumes window to get a complete view of the storage on your system:

1. Choose Devices and Volumes from the Configuration menu of the main window.
2. In the devices and volumes window, choose Show available volumes only from the View menu.
3. Click a volume that is labeled available.
4. Choose Add to domain from the Configuration menu.
5. In the Add Volume dialog box, choose the domain to which the new volume is added.

It is a good idea to balance your domain after you have added a volume. This distributes existing files to the new volume.

D.4.1.4 Removing a Domain

You can delete a domain only after all filesets and fileset clones in the domain are unmounted. You get an error message if you try to remove a domain with mounted filesets. When you remove a domain in order to delete all filesets, its entry in the `/etc/fdmns` directory is removed. However, this can present a security hole since only pointers are changed and no data on the volume is removed. It could be possible to access the fileset data from the command line by using the `salvage` utility (see Section 5.8.4). If you need the increased security, remove each fileset individually.

The deleted domain name remains in the `/etc/fstab` file unless you modify the file by choosing the Modify /etc/fstab option in the dialog box or change it from the command line. If you do not update this file, you will get error messages when you reboot the system or when you enter a `mount -a` command from the command line. If you do not delete the file name from the `/etc/fstab` file, you can do so at a later time from the command line.

To remove a domain:

1. In the object tree, highlight the domain.
2. Choose File Domain from the Configuration menu.
3. Choose Delete from the File Domain menu.
4. In the Delete File Domain dialog box, delete the domain and back up and modify the `/etc/fstab` file.

Note

> When you unmount your filesets and fileset clones, you must choose to modify the `/etc/fstab` file. If you do not do so at that time, you must manually edit the `/etc/fstab` file. After you delete the domain you no longer have access to the filesets.

D.4.1.5 Renaming a Domain

You can assign a new name to an existing domain (see Section 1.6.9). The old name remains in the `/etc/fstab` file unless you modify the file by choosing the Modify /etc/fstab option in the dialog box or change it from the command line. If you do not update the `/etc/fstab` file, the filesets in this domain do not mount when you reboot the system or when you enter the `mount -a` command from the command line. If you must update the `/etc/fstab` file at a later time, do so from the command line.

Unmount all filesets before renaming the domain. You get an error if you try to rename a domain with mounted filesets. You cannot rename a domain with the name of an existing domain.

To rename a domain:

1. In the object tree, highlight the domain.
2. Choose File Domain from the Configuration menu.
3. Choose Rename from the File Domain menu.
4. In the Rename File Domain dialog box, enter the new name and back up and modify the `/etc/fstab` file.

D.4.2 Operations on a Fileset

File system management tasks for filesets are initiated from the main window by selecting Show... or Fileset from the Configuration menu after you have highlighted the fileset on which you want to operate. It is important that you choose to modify and back up the `/etc/fstab` file as you perform fileset operations. If you do not, there is a chance that the `/etc/fstab` file will not be correct for subsequent operations. If you must update the `/etc/fstab` file at a later time, do so from the command line.

D.4.2.1 Creating a Fileset

You cannot create a fileset until you have a domain with which to associate it (see Section D.4.1.1).

To create a fileset:

1. In the object tree, highlight the domain for which the fileset will be created.
2. Choose Fileset from the Configuration menu.
3. Choose New from the Fileset menu.
4. In the New Fileset dialog box, create the fileset, mount the fileset, back up and modify the `/etc/fstab` file, set a free space alert, and set fileset quotas (as desired).

Note that when you create a fileset, you can set the free space alert only as a percentage of the available space.

D.4.2.2 Mounting a Fileset

When you mount a fileset, you make its files available. The display in the main window indicates which filesets are mounted. A domain is active when at least one fileset is mounted.

To mount a fileset:

1. In the object tree, highlight the fileset.
2. Choose Fileset from the Configuration menu.
3. Choose Mount from the Fileset menu.
4. In the Mount Fileset dialog box, create a mount point if it does not exist and back up and modify the `/etc/fstab` file.

D.4.2.3 Unmounting a Fileset

When you unmount a fileset, its files are no longer available. You must unmount the fileset before you can remove it or remove the domain to which it belongs. The display in the main window indicates whether filesets are mounted or not mounted.

To unmount a fileset:

1. In the object tree, highlight the fileset.
2. Choose Fileset from the Configuration menu.
3. Choose Unmount from the Fileset menu.
4. In the Unmount Fileset dialog box, unmount the fileset and back up and modify the `/etc/fstab` file.

D.4.2.4 Setting a Free Space Alert for a Fileset

Set a free space alert to warn when the free space threshold in the fileset is reached or passed. You must mount the fileset before you can set the alert. After you set the alert, you can choose to automatically run a script when the alert threshold is crossed.

To set the free space alert:

1. In the object tree, highlight the fileset.
2. Choose Fileset from the Configuration menu.
3. Choose Set free space alert from the Fileset menu.
4. In the Fileset dialog box, set the free space alert.

You must reset the free space alert when the size of a fileset changes. Free space alert values are evaluated using actual size (for example, kilobytes), not percentage of space.

D.4.2.5 Setting Fileset Quotas

Set fileset quotas to limit the amount of space the fileset can consume. If you do not set quotas, any fileset can use all the available space in the domain. You can only set quotas on mounted filesets.

To set fileset quotas:

1. In the object tree, highlight the fileset.
2. Choose Fileset from the Configuration menu.
3. Choose Set fileset quotas from the Fileset menu.
4. In the Fileset dialog box, set the fileset quotas.

You can view fileset quota information from the main window by choosing Fileset Quotas from the View menu. To return to viewing disk usage, choose the items you want to view from the View menu.

D.4.2.6 Removing a Fileset

You must unmount a fileset before you can delete it. You can unmount as part of the removal process. If the fileset has a clone, you must remove the clone first. Removing a fileset removes all files in that fileset.

To remove a fileset:

1. In the object tree, highlight the fileset.
2. Choose Fileset from the Configuration menu.
3. Choose Delete from the Fileset menu.

4. In the Delete Fileset dialog box, unmount the fileset, delete it, and back up and modify the `/etc/fstab` file.

If all the filesets are unmounted, the fastest way to remove all filesets is to remove the domain to which they belong. However, this might present a security hole because it might be possible to access the data by using the `salvage` utility (see Section 5.8.4).

D.4.2.7 Renaming a Fileset

You must unmount a fileset before you can rename it. You can do this as part of the renaming process. The new fileset name must be unique within the domain.

To rename a fileset:

1. In the object tree, highlight the fileset.
2. Choose Fileset from the Configuration menu.
3. Choose Rename from the Fileset menu.
4. In the Rename Fileset dialog box, unmount the fileset, enter a new name, and back up and modify the `/etc/fstab` file.

Note

If you rename a fileset that has never been mounted (does not already have an `/etc/fstab` file entry) and check the modify `/etc/fstab` box, you do not get a new entry for the fileset name. You must manually edit the `/etc/fstab` file to add the new entry.

If you rename a fileset, its clone can no longer track it. You must delete the old clone and create a new one because clones cannot be renamed.

D.4.3 Operations on an AdvFS Fileset Clone

When you operate on a clone, it is important to modify and back up the `/etc/fstab` file as you perform clone operations. If you do not, there is a chance that the `/etc/fstab` file will not be correct in subsequent operations. If you must update the `/etc/fstab` at a later time, do so from the command line.

D.4.3.1 Creating an AdvFS Fileset Clone

Creating a fileset clone allows you to back up files while the file system is on line (see Section 3.3). You can create only one clone for a fileset.

To create a clone:

1. In the object tree, highlight the fileset that you want to clone.
2. Choose Clone from the Configuration menu.
3. Choose New from the Clone menu.
4. In the New Clone dialog box, enter a clone name, create a mount point, and back up and modify the `/etc/fstab` file.

D.4.3.2 Mounting a Clone

You must mount a clone in order to access it. An unmounted clone tracks changes to a fileset but cannot be read. Clones are mounted as read-only.

To mount a clone:

1. In the object tree, highlight the clone.
2. Choose Clone from the Configuration menu.
3. Choose Mount from the Clone menu.
4. In the Mount Clone dialog box, create a mount point if it does not exist and back up and modify the `/etc/fstab` file.

D.4.3.3 Unmounting a Clone

You cannot access an unmounted clone, but it still tracks fileset changes.

To unmount a clone:

1. In the object tree, highlight the clone.
2. Choose Clone from the Configuration menu.
3. Choose Unmount from the Clone menu.
4. In the Unmount Clone dialog box, unmount the clone and back up and modify the `/etc/fstab` file.

D.4.3.4 Removing a Clone

You must unmount a clone before you can remove it. You can do this as part of the removal process.

To remove a clone:

1. In the object tree, highlight the clone.
2. Choose Clone from the Configuration menu.
3. Choose Delete from the Clone menu.

4. In the Delete Clone dialog box, unmount the clone, delete it, and back up and modify the `/etc/fstab` file.

D.4.4 Operations on a Volume

File system management tasks for volumes are initiated from the main window by selecting Show or Volume from the Configuration menu after you have highlighted the volume on which you want to operate. (In order to view volumes, the View menu must be set to Domains and volumes or Domains, filesets, and volumes.)

D.4.4.1 Setting a Free Space Alert for a Volume

Set a free space alert to inform you when the free space threshold in the volume is reached or passed.

To set the free space alert:

1. In the object tree, highlight the volume.
2. Choose Volume from the Configuration menu.
3. Choose Set free space alert from the Volume menu.
4. In the Volume dialog box, set the free space alert.

When you set the alert, you can choose to automatically run a script when the threshold is crossed.

D.4.4.2 Adding a Volume

Adding a volume is the same as enlarging a domain (see Section D.4.1.3). See Section 1.5 for detailed information about volumes.

D.4.4.3 Removing a Volume

You can remove a volume from a domain at any time without interrupting the logical structure of the filesets in the domain (see Section 1.6.7). The data that was stored on that volume is moved to other volumes in the domain. In order to complete successfully, there must be room on the remaining volumes.

Before you can remove a volume from a domain, all filesets in that domain must be mounted. You cannot remove a volume while you are balancing or defragmenting a domain associated with that volume.

To remove a volume:

1. In the object tree, highlight the volume.
2. Choose Volume from the Configuration menu.
3. Choose Remove from the Volume menu.

4. In the Remove Volume dialog box, confirm the removal.

D.4.5 Maintaining AdvFS

File system maintenance tasks are initiated from the main window through the Maintenance menu. The maintenance tools improve read/write performance by altering the way files are mapped on the disk. They can be run while the system is on line, and their operation is transparent to system users and to applications. The maintenance tasks can be stopped without harm to the domain. The activity that has already taken place remains.

D.4.5.1 Balancing a Multivolume Domain

You can use the GUI to initiate the balance process to evenly distribute files among volumes (see Section 4.10). You cannot balance a domain while you are defragmenting, adding or removing volumes, or removing filesets on the same domain. You can choose to abort the operation at any time. The domain you want to balance must have all filesets mounted. If you try to balance a domain that includes unmounted filesets, you get an error message.

To balance a domain:

1. In the object tree, highlight the domain.
2. Choose Balance from the Maintenance menu.
3. In the Balance dialog box, start the balance process.

D.4.5.2 Defragmenting a Domain

You can make files in a domain more contiguous by defragmenting the domain. Defragmenting also consolidates free space so files created later are also less fragmented (see Section 4.8). You cannot defragment a domain while you are balancing, adding or removing volumes, or removing filesets on the same domain. You can choose to abort the defragment operation at any time. The domain you want to defragment must have all filesets mounted. If you try to defragment a domain that includes unmounted filesets, you get an error message.

To defragment a domain:

1. In the object tree, highlight the domain.
2. Choose Defragment from the Maintenance menu.
3. In the Defragment dialog box, start the defragment process and specify how long it is to operate.

D.5 GUI Operation

Table D–3 presents some GUI problems and solutions. Check dtadvfs(8) and advfsd(8) for additional information.

Table D–3: GUI Troubleshooting

Problem	Cause/Solution
GUI starts slowly	Mount at least one fileset from the command line.
Advanced File System icon not in the Application Manager - Storage Management window	Installation is not complete.
Permission denied message when entering dtadvfs from the command line	You are not the root user.
Password Error dialog when starting dtadvfs from CDE icon	You have entered an incorrect root password. Do not reenter. Cancel the dialog and try again.
Object key is invalid or Can't get the object data error message	GUI sometimes cannot track several actions simultaneously. Restart the GUI.
Slow performance or unexplained error messages appear	Network overload (the GUI cannot get configuration data in a reasonable number of attempts) or system overload (the agent gets few or no CPU cycles). If there are several off-line HSZ devices, unmounted filesets, or spun down disk devices, add them to the disks.ignore file. If none of the above appears to be the problem, the agent is probably hung. Exit the GUI and stop the agent, then restart the agent and the GUI.
Incorrect error message	Configuration tasks are running simultaneously and more than one task failed.
Free space alert value incorrect	Reset. Alert values are evaluated using actual size (for example, in kilobytes), not percentage of space.
Domain deleted but associated volumes shown as in use	Manually change the disk label on the device; use the command line disklabel command or the CDE Disk Configuration utility.
Volume removed but shown	Manually change the disklabel on the device; use the command line disklabel command or the CDE Disk Configuration utility.

Table D–3: GUI Troubleshooting (cont.)

Problem	Cause/Solution
`advfsd` consuming high CPU and I/O resources	Change the agent state monitor interval to reduce the polling frequency (see Section D.2.1). Decrease the number of LSM disk groups and volumes. Decrease the number of unmounted filesets. Inactive domains are particularly slow. Use the `disks.ignore` file (see Section D.2.2.2). If none of the above appears to be the problem the agent is probably hung. Exit the GUI and stop the agent, then restart the agent and the GUI.
System panics at boot time	Check for a bad domain. Panic occurs even if you have removed the domain from the `/etc/fstab` file. Do one of the following in single-user mode, then reboot to multiuser mode: - Repair the domain. - Move the bad domain from the `/etc/fdmns` directory to another directory. (This makes repair difficult.)
Sort gives incorrect results	Check that no data fields to be sorted are marked unavailable. Mount filesets if needed.
`Volume in use` message, when adding a volume	Check that this is not an unrecognized LSM volume (see Section D.2.2.6).
X or Motif problems	Failures related to these can be ignored. Full keyboard support as defined in the Motif style guide is not implemented.

E

Using Controller-Based Cloning and Snapshotting

An AdvFS fileset clone (see Section 1.7.11) is a read-only snapshot of fileset data at the time the clone was created. The fileset clone tracks the changes to the fileset over the life of the clone. AdvFS fileset clones are analogous to controller-based snapshots. Controller-based clones are similar to LSM split mirrors. Controller-based cloning and snapshotting do not require that the AdvFS file system be installed.

Controller-based cloning and snapshotting are hardware operations. You can use controller-based cloning to create a full duplicate data set on different physical storage media from the original data set. Controller-based snapshotting copies only the critical RAID metadata for a data set to new media. It employs copy-on-write technology to preserve a point-in-time copy of the original data set.

You can mount controller-based clones and snapshots as new AdvFS file systems and use the new file systems in back up operations either on the original host or on a different host. If you mount the new file systems on a different host, you can perform backupswithout introducing a performance drain on the source system during the backup operation. You can also use controller-based clones and snapshots as the data sources for decision-support applications.

Caution

Follow the directions in the Best Practice for *Using Controller-Based Cloning and Snapshotting with Tru64 UNIX*. Failure to do so may result in unmountable file systems, data corruption, AdvFS domain panics, or Tru64 UNIX kernel panics. See **`http://www.tru64unix.compaq.com/docs/`** for more information.

Glossary

This glossary defines some of the terms and acronyms used in the AdvFS documentation.

AdvFS Utilities

A layered product available by license. The utilities allow you to create and manipulate multivolume domains.

advfsd

The agent that issues commands and obtains system information for the AdvFS GUI.

agent

The agent runs in the traditional style of the UNIX daemon.

See also *advfsd*

agent state monitor interval

The time interval between agent scans of the system disk.

asynchronous I/O

A nonblocking I/O scheme where data is written to the cache and might return control before the data is written to the disk.

atomic write data logging

Guarantees that all data in a write system call (up to 8 KB) is either written to the disk or none of the data is written to the disk.

balance

To even the distribution of files between volumes of a domain.

bitfile

A set of pages that AdvFS views as one entity. Reserved files and user files are bitfiles.

bitfile metadata table

See *BMT*

block

A 512-byte unit of disk storage. Sixteen blocks comprise a page.

BMT

Bitfile metadata table. An array pages, each with a header and an array of mcells located on each volume. A BMT contains metadata, including file attributes, file extent maps, fileset attributes, and the POSIX file statistics, for all files that have storage on the volume.

buffer cache

The area of memory that contains the blocks of data read from and/or waiting to be written to disk.

checksum

Blocks created during tape backup for error recovery.

clone

See *fileset clone*

Collect

A real-time performance monitoring application.

contiguous

Storage that is physically adjacent on a disk volume.

copy-on-write

The process by which original information is saved in an AdvFS fileset clone when data in the original file is changed.

defragment

To make each file and free space in a domain more contiguous.

direct I/O

An I/O scheme that synchronously reads and writes data from a file without copying it to a cache.

dirty data

Data that has been written by the application, but the file system has cached it in memory so it has not yet been written to disk.

domain

A named pool of storage that contains one or more volumes. Sometimes referred to as file domain.

domain ID

A set of numbers that identify the domain to the system.

domain panic

A condition that prevents further access to the domain when corruption in the domain is detected. AdvFS allows the filesets in the domain to be unmounted after a domain panic.

Domain Version Number

See *DVN*

DVN

Domain version number. A number in the disk metadata that specifies file structure. Domains created with the operating system software Version 5.0 and later contain a DVN of 4, while domains created under earlier versions of the operating system have a DVN of 3.

dtadvfs

The AdvFS Graphical User Interface (GUI).

/etc/fdmns directory

A directory that contains the domain definitions.

/etc/fstab file

A file that identifies file systems that are to be mounted at system reboot.

extent

Contiguous area of disk space allocated to a file. A file might have zero or more extents.

extent map

A table of the size and location of the extents belonging to a file. Simple files have one extent map; striped files have an extent map for every stripe segment.

file extent

See *extent*

file fragment

Created when a file uses only part of the last page of file storage allocated or has a total size of less than 8 KB.

fileset

A hierarchy of directory and files. A fileset represents a mountable portion of the directory hierarchy of the AdvFS file system.

fileset clone

A read-only copy of a fileset that preserves the data and structure of an existing fileset at the time the clone is created. Initially the clone uses very little space. The first time data in a block assigned to the original fileset changes, the original block is preserved in the clone. As more disk blocks change, the clone uses more disk space. The contents of the AdvFS fileset clone can be backed up while the original fileset remains available to users.

fileset ID

A unique identifier that associates the fileset with its domain.

fileset quota

A quota that limits the amount of disk storage that a fileset can consume or the number of files a fileset can contain.

frag file

A file that is used to allocate storage for files or file segments that are less than 8 KB (one page). Using fragments reduces the amount of wasted disk space.

grace period

The period of time a quota's soft limit can be exceeded as long as the hard limit is not exceeded.

GUI refresh interval

The time interval between updates of the GUI window information.

hard limit

The quota limit for disk block usage or number of files that cannot be exceeded.

inode

A numeric file identifier.

LSM

Logical Storage Manager. LSM is a storage management system that provides data redundancy and volume-level striping.

Logical Storage Manager

See *LSM*

mcell

Metadata cells that contain records of file statistics.

metadata

File structure information such as file attributes, extent maps, and fileset attributes.

migrate

To move files from one volume to another within a domain.

mirror

(v.) To maintain identical copies of data on different storage areas. (n.) One of the copies that is maintained.

miscellaneous metadata bitfile

Maps areas of the volume that do not represent AdvFS metadata, such as the disk label and boot blocks.

NetWorker

NetWorker for Tru64 UNIX provides scheduled, online automated backup.

object

A volume, domain, fileset, or fileset clone managed by the AdvFS GUI.

object tree

The AdvFS GUI hierarchical display of objects.

page

An allocation of 8 KB of contiguous disk space (16 blocks).

PAK

Product authorization key. License to access Compaq Computer Corporation software.

Product Authorization Key

See *PAK*

quota file

A file that stores quota limits and keeps track of number of files, disk block usage, and grace period per user ID or per group ID. Fileset quota information is stored within the fileset metadata.

root tag file

A bitfile that defines the location of all filesets in a domain. Each domain has one.

saveset

A collection of blocks created by the `vdump` utility to save AdvFS backup information.

segment

See *stripe segment*

soft limit

The quota value beyond which disk block usage or number of files is allowed only during the grace period.

sparse file

A file whose pages do not all have allocated disk space.

storage area

See *volume*

storage bitmap

Keeps track of allocated disk space on a volume.

stripe

To distribute data across multiple storage areas. AdvFS stripes individual files. LSM and hardware RAID stripe all files.

stripe segment

For AdvFS, the portion of a striped file that resides on a volume. A file striped across four volumes has four stripe segments. Segments can be migrated from one volume to another.

tag

A unique identifier for an AdvFS file within a fileset.

transaction log file

The log file that records changes to metadata before the changes are written to disk. At regular intervals these changes are written to disk.

trashcan

A directory that contains the most recently deleted files from an attached directory. Trashcan directories can be set up by each user for user files.

UBC

Unified Buffer Cache. The dynamically allocated system buffer cache that holds file data and AdvFS metadata.

Unified Buffer Cache

See *UBC*

volume

For AdvFS, anything that behaves like a UNIX block device. This can be a disk, disk partition, or logical volume.

write-ahead logging

The process by which the modifications to the file-structure information are completely written to a transaction log file before the actual changes are written to disk.

Index