

Tru64 UNIX

AdvFS Administration

Part Number: AA-RH96D-TE

September 2002

Product Version: Tru64 UNIX Version 5.1B 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.

© 2002 Hewlett-Packard Company

UNIX® and Motif® are registered trademarks and The Open Group™ is a trademark of The Open Group in the United States and/or other countries. All other product names mentioned herein may be the trademarks of their respective companies.

Confidential computer software. Valid license from Compaq Computer Corporation, a wholly owned subsidiary of Hewlett-Packard Company, required for possession, use, or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

None of Compaq, HP, or any of their subsidiaries shall be liable for technical or editorial errors or omissions contained herein. The information is provided "as is" without warranty of any kind and is subject to change without notice. The warranties for HP or Compaq products are set forth in the express limited warranty statements accompanying such products. Nothing herein should be construed as constituting an additional warranty.

Contents

About This Manual

1 Planning the File System

1.1	What is AdvFS?	1-1
1.2	License Registration	1-2
1.3	File System Design	1-2
1.3.1	Filesets, Domains, and Volumes	1-3
1.3.2	Transaction Log File	1-3
1.3.3	File Storage Allocation	1-4
1.4	Setting Up an AdvFS File System	1-6
1.4.1	Configuring Domains and Filesets	1-6
1.4.2	Organizing Volumes	1-7
1.5	Choosing Storage for AdvFS	1-8

2 Setting Up the File System

2.1	Overview	2-1
2.2	Volumes	2-1
2.2.1	Volume Attributes	2-2
2.3	Domains	2-2
2.3.1	The /etc/fdmns Directory	2-3
2.3.2	Displaying Domain Information	2-3
2.3.3	Creating Domains	2-4
2.3.3.1	Domain Version Numbers	2-5
2.3.3.2	Creating Domains with DVN4	2-6
2.3.3.3	Creating Domains with DVN3	2-7
2.3.3.4	Upgrading Domains to DVN4	2-7
2.3.4	Increasing the Size of Domains	2-8
2.3.4.1	Increasing Storage in Domains by Adding Volumes ...	2-8
2.3.4.2	Increasing Storage in Domains by Exchanging Volumes	2-11
2.3.4.3	Increasing Storage in Domains by Extending an Existing Volume	2-11
2.3.5	Reducing the Size of Domains	2-12
2.3.6	Removing Domains	2-14
2.3.7	Renaming Domains	2-14

2.4	Filesets	2-15
2.4.1	Designating Filesets in the /etc/fstab File	2-16
2.4.2	Displaying Fileset Information	2-17
2.4.3	Creating Filesets	2-18
2.4.4	Upgrading Filesets	2-18
2.4.5	Mounting Filesets	2-19
2.4.6	Unmounting Filesets	2-20
2.4.7	Removing a Fileset	2-20
2.4.8	Renaming Filesets	2-21
2.4.9	Changing the Characteristics of Filesets	2-21
2.4.10	AdvFS Fileset Clones	2-22
2.4.10.1	Creating an AdvFS Fileset Clone	2-23
2.4.10.2	Mounting and Unmounting an AdvFS Fileset Clone ..	2-24
2.4.10.3	Removing an AdvFS Fileset Clone	2-24
2.4.10.4	Renaming an AdvFS Fileset Clone	2-24
2.5	Configuring an AdvFS root File System	2-24
2.5.1	Mounting the root File System in Single-User Mode	2-25
2.5.2	Changing the Name of the root Domain	2-25
2.5.3	Changing the Name of the root Fileset	2-25
2.6	Setting Up Trashcans for File Recovery	2-27

3 Managing Quotas

3.1	Introducing Quotas	3-1
3.1.1	Quota Limits	3-2
3.1.2	Grace Period	3-3
3.1.3	Summarizing Quota Characteristics	3-4
3.2	User and Group Quotas	3-4
3.2.1	User and Group Quota Files	3-4
3.2.2	Setting User and Group Quotas and Grace Periods	3-5
3.2.2.1	Setting User Quotas	3-6
3.2.2.2	Setting the User Grace Period	3-7
3.2.2.3	Setting Group Quotas	3-8
3.2.2.4	Setting the Group Grace Period	3-9
3.2.2.5	Setting a Temporary Grace Period	3-10
3.2.3	Setting User and Group Quotas for Multiple Users and Groups	3-10
3.2.3.1	Setting Quotas for Multiple Users	3-11
3.2.3.2	Setting Quotas for Multiple Groups	3-11
3.2.4	Activating User and Group Quotas at System Start-Up ...	3-12
3.2.5	Activating User and Group Quotas Manually	3-13
3.2.6	Deactivating User and Group Quotas	3-14
3.3	Fileset Quotas	3-14

3.3.1	Fileset Quota Files	3-14
3.3.2	Setting Fileset Quotas and Grace Periods	3-14
3.3.2.1	Setting Fileset Quotas	3-15
3.3.2.2	Setting the Grace Period	3-16
3.3.3	Setting Fileset Quotas for Multiple Filesets	3-16
3.3.4	Activating Fileset Quotas	3-16
3.3.5	Deactivating Fileset Quotas	3-16
3.4	Verifying Fileset and Disk Space Usage	3-17
3.4.1	Examining User and Group Quotas	3-17
3.4.1.1	Printing the Tag and Full Pathname for Each File	3-17
3.4.1.2	Summarizing Fileset Ownership	3-17
3.4.1.3	Displaying Disk Usage and Limits	3-18
3.4.1.4	Verifying User and Group Quota Consistency	3-19
3.4.1.5	Summarizing User and Group Quotas by Fileset	3-19
3.4.2	Examining Fileset Quotas	3-21
3.4.2.1	Displaying Fileset Space	3-21
3.4.2.2	Displaying Fileset Limits	3-22
3.4.2.3	Displaying Domain Space	3-22
3.4.2.4	Displaying Domain and Fileset Space	3-23

4 Backing Up and Restoring Data

4.1	Data and Metadata Integrity	4-1
4.1.1	Ensuring Data Integrity	4-1
4.1.1.1	Ensuring Data Integrity Using An Application's Quiesce Function	4-2
4.1.1.2	Ensuring Data Integrity Using Synchronous Writing .	4-2
4.1.1.3	Ensuring Data Integrity Without Using Synchronous Writing	4-2
4.1.2	Ensuring Metadata Integrity	4-3
4.1.2.1	Ensuring Metadata Integrity by Freezing Domains ...	4-3
4.1.2.2	Ensuring Metadata Integrity by Unmounting Filesets	4-3
4.2	Creating a Backup Source	4-4
4.2.1	Using a Mounted Original Fileset as a Backup Source	4-4
4.2.1.1	Advantages and Disadvantages	4-4
4.2.1.2	Ensuring Data and Metadata Integrity	4-4
4.2.1.3	Preparing the Backup Source	4-4
4.2.2	Using a Mounted Fileset Clone as a Backup Source	4-5
4.2.2.1	Advantages and Disadvantages	4-5
4.2.2.2	Ensuring Data and Metadata Integrity	4-5
4.2.2.3	Preparing the Backup Source	4-5

4.2.3	Using a Mounted Fileset from an LSM Split Mirror as a Backup Source	4-6
4.2.3.1	Advantages and Disadvantages	4-6
4.2.3.2	Ensuring Data and Metadata Integrity	4-7
4.2.3.3	Preparing the Backup Source	4-7
4.2.4	Using a Mounted Fileset from a Controller-Based Clone as a Backup Source	4-8
4.2.4.1	Advantages and Disadvantages	4-8
4.2.4.2	Ensuring Data and Metadata Integrity	4-8
4.2.4.3	Preparing the Backup Source	4-8
4.2.5	Using a Mounted Fileset from a Controller-Based Snapshot as a Backup Source	4-10
4.2.5.1	Advantages and Disadvantages	4-10
4.2.5.2	Ensuring Data and Metadata Integrity	4-10
4.2.5.3	Preparing the Backup Source	4-10
4.3	Backup Methods and Tools	4-12
4.3.1	Introduction to the vdump and vrestore Commands	4-12
4.3.2	Unique Features of the vdump Command	4-13
4.3.3	Specifying Backup Level Using the vdump Command	4-14
4.3.4	Dumping to Tape Using the vdump Command	4-14
4.3.5	Dumping to Standard Output Using the vdump Command	4-15
4.3.6	Dumping Subdirectories Using the vdump Command	4-15
4.3.7	Dumping to Files or Disk Partitions	4-15
4.3.8	Compressing Savesets	4-16
4.3.9	Dumping with Error Protection	4-16
4.3.10	Listing Saved Files in a vdump Saveset	4-17
4.3.11	Backing Up a System Disk Using the clonefsset and vdump Commands— an Example	4-17
4.3.12	Dumping and Restoring Files Remotely	4-19
4.3.13	Unique Features of the vrestore Command	4-19
4.3.14	Restoring Files Using the vrestore Command	4-20
4.3.15	Restoring Quotas Using the vrestore Command	4-20
4.3.16	Restoring from a Tape with Many Savesets Using the vrestore Command	4-20

5 Optimizing and Tuning the AdvFS File System

5.1	Monitoring Performance	5-2
5.2	Improving Performance by Disabling the Frag File	5-2
5.3	Improving Transaction Log File Performance	5-3
5.3.1	Moving the Transaction Log File	5-4
5.3.2	Isolating the Transaction Log File	5-5

5.4	Data Cache Tuning	5-6
5.5	Improving Data Consistency	5-7
5.5.1	Controlling I/O	5-7
5.5.1.1	Asynchronous I/O	5-7
5.5.1.2	Synchronous I/O	5-7
5.5.1.3	Turning Synchronous I/O On and Off	5-8
5.5.2	Enabling Atomic-Write Data Logging I/O	5-8
5.5.2.1	Asynchronous Atomic-Write Data Logging I/O	5-9
5.5.2.2	Synchronous Atomic-Write Data Logging I/O	5-9
5.5.2.3	Persistent Atomic-Write Data Logging	5-10
5.5.2.4	Temporary Atomic-Write Data Logging	5-10
5.5.2.5	Turning Atomic-Write Data Logging On and Off	5-11
5.6	Improving Data Transfer Rate with Direct I/O	5-11
5.7	Changing Attributes to Improve System Performance	5-12
5.8	Improving Operating System Throughput with the vfast Utility	5-13
5.8.1	Choosing to Run the vfast Utility	5-16
5.8.2	Examples of the vfast Utility	5-17
5.9	Defragmenting a Domain	5-18
5.10	Defragmenting a File	5-22
5.11	Balancing a Multivolume Domain	5-22
5.12	Migrating Files to Different Volumes	5-23
5.13	Striping Files	5-25
5.14	Moving a Domain and its Filesets to a New Volume	5-26
5.15	Controlling Domain Panic Information	5-27

6 Troubleshooting

6.1	Recovering a User File	6-1
6.2	General Recovery Procedures	6-1
6.2.1	Saving Data for a Problem Report	6-2
6.2.2	Saving Copies of System Metadata	6-2
6.2.3	Saving Undamaged Filesets	6-2
6.2.4	Verifying File System Consistency	6-2
6.2.5	Fixing On-Disk Metadata Corruptions with the fixfdmn Utility	6-4
6.2.6	Salvaging File Data from a Corrupted Domain	6-5
6.2.6.1	Recovering File Data	6-6
6.2.6.2	Recovering Data from a Corrupted root Domain	6-7
6.2.6.3	Recovering Data Block by Block	6-8
6.3	Fixing Problems	6-8
6.3.1	Recovering from a Domain Panic	6-8

6.3.2	Recovering from Filesets That are Mounted Read-Only ...	6-10
6.3.3	Restoring the /etc/fdmns Directory	6-12
6.3.3.1	Reconstructing the /etc/fdmns Directory Manually	6-12
6.3.3.2	Reconstructing the /etc/fdmns Directory Using the advscan Command	6-13
6.3.4	Recovering from Data Corruption of an AdvFS root Domain	6-17
6.3.5	Recovering from Hardware Corruption	6-21
6.3.6	Moving an AdvFS Disk to Another Machine	6-22
6.3.7	Restoring a Multivolume usr Domain	6-24
6.3.8	Recovering from Accidental Use of the mkfdmn or addvol Command	6-26
6.4	Preventing Problems	6-26
6.4.1	Checking Free Space and Disk Usage	6-27
6.4.2	Reusing Volumes	6-28
6.4.3	Failing Disks	6-28
6.4.4	Controlling Disk Usage	6-28
6.4.5	Quota and Grace Period Limits	6-29
6.4.5.1	Exceeding Quota Limits	6-29
6.4.5.2	Saving Up to Quota Limits	6-29
6.4.5.3	Changing Quota Limits	6-29
6.4.5.4	Turning Off the Grace Period	6-29
6.4.6	Avoiding Disk Structure Incompatibility	6-29
6.4.7	Avoiding Utility Incompatibility	6-30
6.4.8	Avoiding Log File Inconsistency	6-30
6.4.9	Increasing the Size of an AdvFS root Domain	6-31
6.4.9.1	Increasing the Size of the root Volume	6-31
6.4.9.2	Replacing the root Volume with a Larger Device	6-31
6.4.10	Memory Mapping, Direct I/O, and Data Logging Incompatibility	6-33
6.4.11	Preventing Access to Old Data	6-34
6.4.12	Invalid or Corrupt Saveset Format	6-34
6.4.13	Overcoming Poor Performance	6-34
6.4.14	The rmvol or migrate Command Will Not Start	6-35

A Accessing File System Information with the SysMan Menu

B AdvFS Commands

B.1	AdvFS Base System Commands	B-1
B.2	AdvFS Utilities Commands	B-3
B.3	Comparison of AdvFS and UFS Commands	B-4

C Converting File Systems

C.1	Converting a /usr File System to AdvFS	C-1
C.1.1	Using a Backup Tape	C-1
C.1.2	Using an Intermediate File	C-3
C.1.3	Using a Second Disk	C-4
C.2	Converting the root File System to AdvFS	C-5
C.3	Converting a File System to AdvFS	C-7
C.3.1	Using a Backup Tape	C-7
C.3.2	Using an Intermediate File	C-9
C.3.3	Using a Second Disk	C-10
C.3.4	Using a Second System	C-11
C.4	Converting from AdvFS to UFS	C-13
C.4.1	Converting the root File System to UFS	C-13
C.4.2	Converting a Fileset to UFS	C-14

D Data Management Applications Programming Interface (DMAPI)

E Graphical User Interfaces

E.1	Installing the AdvFS GUI	E-1
E.2	Components of the GUI	E-2
E.2.1	GUI Agent (advfsd)	E-2
E.2.2	AdvFS GUI (dtadvfs)	E-3
E.2.2.1	GUI Security	E-3
E.2.2.2	Ignoring Disks	E-4
E.2.2.3	Log Files	E-4
E.2.2.4	Help	E-5
E.2.2.5	Monitoring Remote File Systems	E-5
E.2.2.6	Managing LSM Volumes	E-5
E.3	Displaying File System Information	E-6
E.3.1	Choosing Units for Displaying Information	E-7
E.3.2	Main Window	E-7
E.3.3	Devices and Volumes Window	E-8
E.4	Performing File System Operations	E-8
E.4.1	Operations on a Domain	E-8
E.4.1.1	Creating a Domain	E-8
E.4.1.2	Setting a Free Space Alert for a Domain	E-9
E.4.1.3	Enlarging a Domain	E-9
E.4.1.4	Removing a Domain	E-10
E.4.1.5	Renaming a Domain	E-11
E.4.2	Operations on a Fileset	E-11

E.4.2.1	Creating a Fileset	E-11
E.4.2.2	Mounting a Fileset	E-12
E.4.2.3	Unmounting a Fileset	E-12
E.4.2.4	Setting a Free Space Alert for a Fileset	E-12
E.4.2.5	Setting Fileset Quotas	E-13
E.4.2.6	Removing a Fileset	E-13
E.4.2.7	Renaming a Fileset	E-14
E.4.3	Operations on an AdvFS Fileset Clone	E-14
E.4.3.1	Creating an AdvFS Fileset Clone	E-14
E.4.3.2	Mounting a Clone	E-15
E.4.3.3	Unmounting a Clone	E-15
E.4.3.4	Removing a Clone	E-15
E.4.4	Operations on a Volume	E-15
E.4.4.1	Setting a Free Space Alert for a Volume	E-16
E.4.4.2	Adding a Volume	E-16
E.4.4.3	Removing a Volume	E-16
E.4.5	Maintaining AdvFS	E-17
E.4.5.1	Balancing a Multivolume Domain	E-17
E.4.5.2	Defragmenting a Domain	E-17
E.5	GUI Operation	E-18

F Application Programming Interface (API)

Glossary

Index

Figures

1-1	AdvFS File System Design	1-3
2-1	Single-Volume Domain	2-5
2-2	Enlarging a Domain	2-9
2-3	Cloning a Fileset	2-22
5-1	Balancing a Domain	5-16
5-2	Migrating Files	5-24

Tables

1-1	Domain and Fileset Configuration Tradeoffs	1-7
1-2	AdvFS Volume Configuration Tradeoffs	1-8
2-1	Trashcan Commands	2-28
3-1	Quota Size Limits	3-2

3-2	Comparing User and Group Quotas with Fileset Quotas	3-4
3-3	User and Group Usage Commands	3-17
3-4	User and Group Usage Commands	3-21
4-1	Using a Mounted Original Fileset as a Backup Source	4-4
4-2	Using a Mounted Fileset Clone as a Backup Source	4-5
4-3	Using an LSM Split Mirror as a Backup Source	4-6
4-4	Using a Mounted Fileset from a Controller-Based Clone as a Backup Source	4-8
4-5	Using a Mounted Fileset from a Controller-Based Snapshot as a Backup Source	4-10
5-1	Turning Atomic-Write Data Logging On	5-11
5-2	Turning Atomic- Write Data Logging Off	5-11
6-1	Incorrect Number of Links	6-14
6-2	Incorrect Domain Volume Count	6-15
6-3	Incorrect Number of Partitions	6-15
6-4	Disk Space Usage Commands	6-27
B-1	AdvFS Configuration Commands	B-1
B-2	AdvFS Information Display Commands	B-1
B-3	AdvFS Backup Commands	B-2
B-4	AdvFS Check and Repair Commands	B-2
B-5	AdvFS Quota Commands	B-2
B-6	AdvFS Disk Structure Commands	B-3
B-7	AdvFS Utilities Commands	B-3
B-8	Comparison of AdvFS and UFS Commands	B-4
E-1	AdvFS GUI Subsets	E-1
E-2	Disk Space Requirements for the AdvFS GUI	E-2
E-3	GUI Troubleshooting	E-18

About This Manual

This manual provides task-based information and instruction for understanding, configuring, troubleshooting, and tuning the Advanced File System (AdvFS). AdvFS is the default file system for the HP Tru64 UNIX operating system.

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 the *Logical Storage Manager* manual.) The AdvFS GUI, which runs under the Common Desktop Environment (CDE), lets you perform AdvFS operations in a graphical environment. (See *Appendix E*.)

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

A description of the following new features added to AdvFS in Version 5.1B are included in this manual:

- File system quiesce capability with new `freezefs` and `thawfs` commands (*Section 4.1.2.1*)
- A background optimization tool, `vfast` (*Section 5.8*)
- Dynamic volume expansion using the `mount` command options (*Section 2.3.4.3*)

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

- A domain recovery tool, `/sbin/advfs/fixfdmn`, that examines metadata corruption (*Section 6.2.5*)
- A `mount` option for temporary atomic-write data logging for a file (*Section 5.5*)
- 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 (*Section 2.4.9*)

- The ability to turn frag files on and off by using the `mkfset` and `chfsets` commands (*Section 5.2*)
- AdvFS cache tuning that is done through the UBC parameters (*Section 5.4*)
- Improved disk structure that increases the number of files the domain can track (*Section 2.3.3*)
- A disk salvage utility that can recover information at the block level from disks that have been damaged (*Section 6.2.6*)
- An improved directory structure that increases the speed of file creation and access (*Section 2.3.3.1*)
- Enhanced `vdump` and `vrestore` command capability (*Chapter 4*)
- Remote device support for backup and restore (*Chapter 4*)
- Increased quota limits (*Chapter 3*)
- Direct I/O to allow unbuffered, synchronous I/O (*Section 5.6*)
- Smooth sync option to promote continuous I/O (*Section 5.7*)
- A disk usage display tool, the `/sbin/advfs/vdf` utility (*Section 3.4.2.4*)
- Metadata display utilities (*Section 6.2.2*)
- A data management API (DMAPI) (*Appendix D*)

Organization

This manual is organized as follows:

<i>Chapter 1</i>	Describes the volumes, domains, and filesets that make up the AdvFS file system
<i>Chapter 2</i>	Explains how to set up your system
<i>Chapter 3</i>	Explains user and group quotas as well as fileset quotas, which are unique to the AdvFS file system
<i>Chapter 4</i>	Provides directions for creating a backup source and backing up and restoring data
<i>Chapter 5</i>	Provides guidelines for optimizing and tuning your AdvFS file system
<i>Chapter 6</i>	Provides information to help you troubleshoot problems with your AdvFS file system
<i>Appendix A</i>	Explains the use of the SysMan menu
<i>Appendix B</i>	Lists all the AdvFS commands and compares them with UFS commands

<i>Appendix C</i>	Provides an explanation and sample code for converting a file system to AdvFS
<i>Appendix D</i>	Describes the data management applications programming interface (DMAPI)
<i>Appendix E</i>	Describes the AdvFS Graphical User Interface
<i>Appendix F</i>	Lists the Application Program Interface (API) commands.

Related Documentation

Other useful Tru64 UNIX documentation includes:

- The *AdvFS Quick Reference* guide, which provides an overview of AdvFS, is available on the World Wide Web at the following URL:
`http://www.tru64unix.compaq.com/docs/techtips.html`
- The *System Administration* manual describes the tasks you perform to administer the operating system running on a workstation or server.
- The *Hardware Management* manual describes the tasks you perform to administer hardware that is running the Tru64 UNIX operating system.
- The *Installation Guide* describes how to install your operating system software.
- The *Logical Storage Manager* manual provides information on storage management.
- The *Programmer's Guide* describes the development environment of the Tru64 UNIX operating system.
- The *Technical Overview* manual provides a brief overview of Tru64 UNIX and its components.
- The *Software License Management* manual describes how to use the License Management Facility (LMF) to manage software licenses on a Tru64 UNIX system.
- The *System Configuration and Tuning* manual provides information on system performance tuning and advanced kernel configuration.
- The *Cluster Administration* manual describes how to manage systems that run TruCluster software.
- Best Practice documentation illustrates the optimal way to perform common administrative tasks.
- Storage Area Network information can be found at `http://www.compaq.com/storage/software_index.html`

- The *System Limits* manual, *Release Notes*, and *QuickSpecs* for Tru64 UNIX and AdvFS Utilities 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/>

It is also available on the documentation CD-ROM.

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 HP.) 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.

The web page <http://www.tru64unix.compaq.com/docs/> provides information on all of the books in the Tru64 UNIX documentation set.

Reader's Comments

HP 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 HP technical support office. Information provided with the software media explains how to send problem reports to HP.

Conventions

This guide uses the following conventions:

#	A number sign represents the superuser prompt.
% cat	Boldface type in interactive examples indicates typed user input.
<i>file</i>	Italic (slanted) type indicates variable values, place holders, 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, <code>cat(1)</code> indicates that you can find information on the <code>cat</code> 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.

Planning the File System

The Advanced File System (AdvFS) is the default file system for the HP Tru64 UNIX operating system. This chapter discusses the following topics:

- Comparing AdvFS with UFS (Section 1.1)
- License registration (Section 1.2)
- Design and components of an AdvFS file system (Section 1.3)
- Configuration of the file system (Section 1.4)
- Storage alternatives for AdvFS (Section 1.5)

1.1 What is AdvFS?

The AdvFS file system differs from the traditional UNIX File System (UFS). With AdvFS you can modify your system configuration at any time without shutting down the system. Because AdvFS with AdvFS Utilities supports a multivolume file system, you can easily add or remove storage as your system requirements change. In addition, **Logical Storage Manager (LSM)** volumes and storage area networks (SANs) can be used for AdvFS storage.

In contrast, the UFS model is rigid. Each disk (or disk partition) contains a single file system. 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.

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.

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

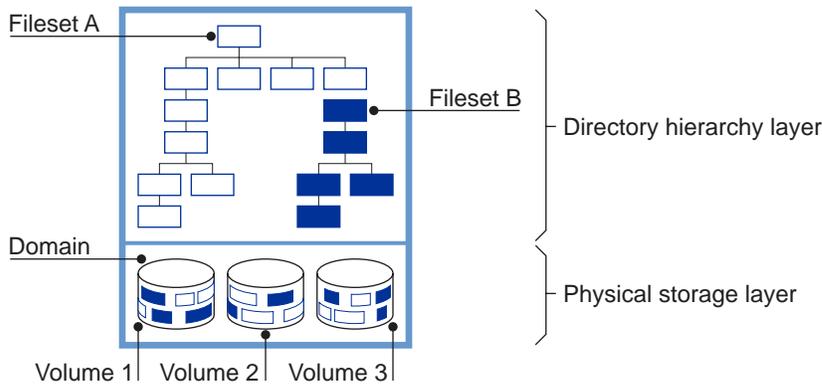
The AdvFS file system consists of two distinct layers: the directory hierarchy layer and the physical storage layer. The directory hierarchy layer implements the file-naming scheme and POSIX-compliant functions such as creating and opening files, or reading and writing files. The physical storage layer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

This decoupled file system structure enables you to manage the physical storage layer apart from the directory hierarchy layer. 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 on the physical level is completely transparent to end users.

AdvFS is the file system for the cluster configuration. Cluster operation is likewise transparent. AdvFS running on a cluster, with very few exceptions, looks no different from AdvFS running on a single node.

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



For the types of storage that AdvFS can use, see Section 1.5.

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 1.4.1 for information about configuring an AdvFS file system and Section 2.4 for information about fileset operation.

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 2.3 for information about domain management.

A **volume** is any mechanism that behaves like a UNIX block device. (See Section 1.4.2.) When first created, all domains consist of a single volume. If you have the optional AdvFS Utilities, you can transform a single-volume domain into a multivolume domain by adding one or more volumes to it.

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. 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 5.5.) 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 as follows:

- Extents

An **extent** is a contiguous area of disk space that AdvFS allocates to a file. When storage is added to a file, it is grouped in extents. An **extent map** exists 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 than a similar-sized file with a few large extents.

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

- Disk fragmentation

When a disk is highly fragmented, there are many small free spaces. AdvFS writes data to isolated physical pages because contiguous pages are not available. This results 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 `vfast` utility to optimize file placement. (See Section 5.8.)

- 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 pages. In files where holding the last bytes of a page would represent a waste of more than 5% of the allocated space, a **file fragment** is created. The fragment 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. If disk space usage is not a consideration, you can disable the frag file. (See Section 5.2.)

- 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(2)` command or by using the `lseek(2)` and `write(1)` 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` command with the `-l` option. The `ls` command with the `-s` option displays the amount of storage actually used by the file. Section 6.4.1 details a number of ways to look at disk usage.

1.4 Setting Up an AdvFS File System

When planning your configuration, consider setting up the root (/), /usr and /var 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 file system, which is restricted to one volume) to increase the size of the domain without changing anything in the existing configuration.

1.4.1 Configuring Domains and Filesets

You can create one domain that contains numerous filesets or you can create a domain for each fileset; you can also structure your system anywhere between these two extremes. If you have AdvFS Utilities, you can distribute your domains on more than one volume.

If you have enough drives, use an entire disk for a domain. Add one partition (typically, partition c) to a domain rather than several partitions on one disk (such as a, b, g, h).

Do not place partitions in different domains, for example, partition a in one domain and partition b in another domain.

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.

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 can create domains for each of your organization's departments, such as engineering, finance, and personnel.

A data management applications programming interface (DMAPI)-enabled domain (Appendix D) can have only one fileset.

Table 1–1 illustrates the advantages and disadvantages of choosing different configurations.

Table 1–1: Domain and Fileset Configuration Tradeoffs

Configuration	Advantage	Disadvantage
Entire disk used for a domain, not small partitions on several physical devices	Efficiency; device's I/O stream is not shared	Waste of space if small domains are on large volumes
Domain configured on multiple volumes (requires Advanced Utilities) rather than on a single volume	Distributes I/O load; increases throughput	Failure of a single volume makes the domain inaccessible
Many smaller domains instead of one large domain (assuming the number of volumes stays the same)	Faster recovery; improved throughput; might decrease some administrative tasks	Might increase administrative overhead
Many filesets in a domain instead of one large one	Can set quota limits on each fileset for finer-grained control of resources	Administrative overhead; all filesets must be mounted to run some utilities
Transaction log on fast or uncongested volume	Prevents log from becoming I/O bottleneck	Cost of hardware
Root fileset configured as AdvFS	Fast recovery after system crash	None

You can examine your file system in the following ways:

- Use the `showfssets` command to display the number of filesets in a domain and the size of the filesets.
- Use the `showfile` command to determine which volume a file resides on. Make sure that busy files are not located on the same volume.

Chapter 5 details ways to optimize a file system. The *System Configuration and Tuning* guide provides detailed guidelines for planning and configuring your file system. See Appendix E if you want to configure your domains using a graphical user interface.

1.4.2 Organizing Volumes

A volume is the entity that is a block special device (such as a single disk partition, an entire disk, an aggregate volume provided by LSM, hardware RAID, or a SAN). Each volume can be assigned to only one domain. It is associated with its domain by a **domain ID**, which is automatically stored in the in the AdvFS metadata on the volume. If you have the optional AdvFS Utilities, you can add more than one volume to a domain.

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

The following table describes the tradeoffs for how you configure domains on AdvFS volumes.

Table 1–2: AdvFS Volume Configuration Tradeoffs

Configuration	Advantage	Disadvantage
Volume is a disk partition	Uses small portion of disk	Limited size; I/O contention with other partitions
Volume is a single disk	More efficient than using single partitions	Limited size
Volume is a RAID volume (LSM or hardware) or SAN	Improve performance and reliability; increased capacity	Cost: requires LSM license or additional hardware

1.5 Choosing Storage for AdvFS

When you create a domain, AdvFS recognizes storage through the block special device name that is associated with the domain. Storage can include directly connected disks, software RAID volumes, hardware RAID, and storage area networks.

By using LSM, hardware RAID and SAN volumes, you can take advantage of their features that allow for advanced backup strategies and dynamic volume expansion. For more information on LSM see the *Logical Storage Manager* manual. For information on hardware RAID and SAN products see <http://www.tru64unix.compaq.com/storage/index.html>.

Setting Up the File System

Once you have determined the configuration of your file system, you can create and structure it. This chapter discusses the following topics:

- Setting up your AdvFS file system (Section 2.1)
- Volumes (Section 2.2)
- Domains (Section 2.3)
- Filesets (Section 2.4)
- Configuring an AdvFS root file system (Section 2.5)
- Setting up your system to make deleted files retrievable (Section 2.6)

2.1 Overview

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

To create an active domain, do the following:

1. Create a domain and assign a volume to it (Section 2.3.3).
2. Create a fileset (Section 2.4.3).
3. Create a mount-point directory (Section 2.4.5).
4. Mount a fileset (Section 2.4.5).

You can create one fileset per domain or you can create many. See Section 1.4.1 for guidelines.

AdvFS is fully supported in the `/etc/fstab` file to automatically mount a fileset at system boot. (See Section 2.4.1.) Alternately, you can use a graphical interface (Appendix E) or the SysMan Menu (Appendix A) to perform this activity.

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

2.2 Volumes

An AdvFS volume can be a raw disk partition, an entire disk, an aggregate volume provided by Logical Storage Manager (LSM), a storage area network

(SAN), or a hardware or software redundant array of independent disks (RAID) storage.

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.

See Section 2.3 for information on how to associate volumes with domains.

2.2.1 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.

To display or modify the current volume attributes, use the SysMan Menu utility called Manage an AdvFS Domain (see Appendix A), or enter the `chvol` command from the command line:

```
chvol device_name domain_name
```

The `chvol` command supports shorthand names for LSM volume names. The following commands are equivalent:

```
# chvol dom1 testdg.vol1  
# chvol dom1 /dev/vol/testdg/vol1
```

You can change the attributes of a volume in an active domain at any time; the system does not have to be quiescent. See Chapter 5 and the *System Configuration and Tuning* manual for more complete information on modifying attributes to improve system performance.

2.3 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 (Section 2.3.1), 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.

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.

2.3.1 The `/etc/fdmns` Directory

The `/etc/fdmns` directory defines domains by providing a subdirectory for each domain you create. The domain name is defined by a directory entry. When you create a domain, the directory entry is created. 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 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.

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 metadata information. In addition, each domain is defined by an entry in the `/etc/fdmns` directory. (See Section 2.3.1.) 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.

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 `/sbin/advfs/advscan` utility to locate the volumes and reconstruct the directory. (See Section 6.3.3.)

2.3.2 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 22 12:33 2002 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%
```

To examine disk usage and quota values, see Section 3.4.

2.3.3 Creating Domains

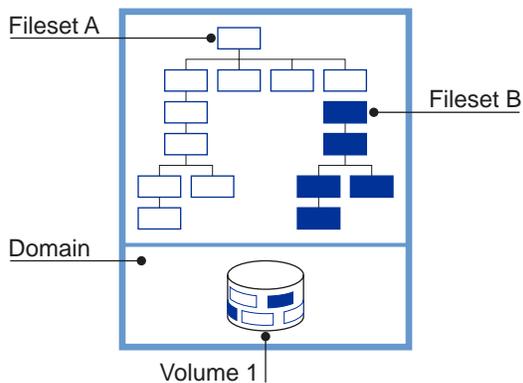
The first step in setting up an AdvFS file system is creating a domain and assigning an initial volume to it. However, a domain is not a complete file system that you can mount. To mount an AdvFS file system, the domain must contain one or more filesets. You can access files as soon as you mount one or more filesets. (See Section 2.4.) For the advantages and disadvantages of different domain and fileset configurations, see Section 1.4.1.

To convert a UFS file system to AdvFS or to configure the root domain as AdvFS, see Appendix C.

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 2-1, use the SysMan Menu utility called Create a New AdvFS Domain (see Appendix A), a graphical user interface (see Appendix E), or enter the `mkfdmn` command from the command line:

mkfdmn *volume_name domain_name*

Figure 2–1: Single-Volume Domain



If you create a domain on a partition and it overlaps with a partition that has an entry in the `/etc/fdms` directory (that is, the partition is associated with a domain), you get an error message. You can override the message. Alternatively you can execute the `mkfdmn` command with the `-F` option to bypass this test.

If you try to create a domain on a volume that is marked in the disk label as in use, the `mkfdmn` command fails if the volume is mounted or if it is a current swap partition. If the volume is unmounted, you get a 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/disk/dsk3g is marked in use for 4.2BSD.
If you continue with the operation you can
possibly destroy existing data.
CONTINUE? [y/n] <y>
```

The `mkfdmn` command supports shorthand names for LSM volume names. The following commands are equivalent:

```
# mkfdmn dom1 testdg.vol1
# mkfdmn dom1 /dev/vol/testdg/vol1
```

2.3.3.1 Domain Version Numbers

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.

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.

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 5.0 or later software. Therefore, if your application requires backward compatibility, do not upgrade your domain.

DVN4 domains have an improved disk structure that provides support for quota values larger than two terabytes and increases performance for directories containing thousands of files. DVN4 domains automatically have an index created when the directory grows beyond a page, that is, about 200 files. Directories with more than 5000 files show the most benefit. Execute 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.

2.3.3.2 Creating Domains 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 Version 4 of the operating system, all existing domains retain the DVN of 3. Thus `/root`, `/usr`, and `/var` also have a DVN of 3. No conversion utility exists. To move your data to a DVN4 domain, see Section 2.3.3.4.

To create an active single-volume DVN4 domain, do the following:

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 a 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.

Caution

Do not use the `mkfdmn` command on a volume containing data that you want to keep. Executing the command destroys the data on that volume. If you have accidentally used the `mkfdmn` command, you might recover some of your data by using the

`/sbin/advfs/salvage` utility if the volume is an AdvFS volume. (See Section 6.2.6.)

The following example creates a single-volume DVN4 domain, `domain_2`, and two filesets, `fileset_a` and `fileset_b`, in the domain. Because the domain has only one volume, the files in both filesets physically reside on that volume. 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 fileset_a
# mkfset domain_2 fileset_b
# mkdir /fileset_a
# mkdir /fileset_b
# mount domain_2#fileset_a /fileset_a
# mount domain_2#fileset_b /fileset_b
```

You can set up a domain with various kinds of storage. See Section 1.5 for more information. 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 a non-cluster root domain) into a multivolume domain. Use the SysMan Menu utility called Manage an AdvFS Domain (see Appendix A), the AdvFS GUI (see Appendix E), or see Section 2.3.4.

2.3.3.3 Creating Domains with DVN3

You can create a DVN3 domain on a Version 5 system. Execute the `mkfdmn` command with the `-v3` option. This is useful if you need backward compatibility.

For more information about creating a DVN3 domain, see `mkfdmn(8)`. See Section 2.4.5 for directions on mounting the DVN3 domain you have created.

2.3.3.4 Upgrading Domains to DVN4

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 2.3.3 first. To upgrade a domain, do the following:

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. Unmount the old domain and remove it 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 BMT allocations.

4. Create the new filesets by using the `mkfset` command and mount them.
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 `fileset_p` and `fileset_m` and put them back on the same volume with the same names, enter the following commands:

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

2.3.4 Increasing the Size of Domains

You can expand a domain by doing any of the following:

- Adding another volume to the domain (requires AdvFS Utilities).
- Replacing a volume in the domain with a larger storage device.
- Increasing the size of the underlying storage and notifying the domain.

Neither increasing nor decreasing size affects the directory hierarchy layer; all pathnames for the files remain the same. The file system can remain active during the process.

Except in a cluster configuration, the root domain is restricted to one volume. If you need to increase the size of your root domain, depending on your volume, you can increase the underlying storage and notify the domain, or you can follow the procedure in Section 6.4.9 to change the device on which root is located.

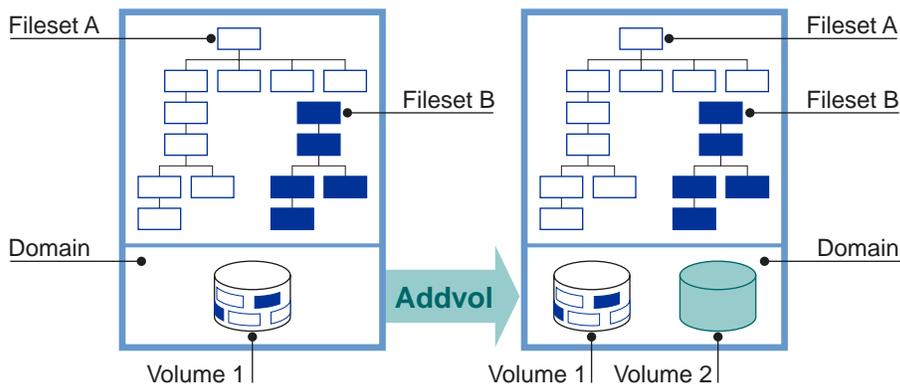
Unless quotas are set, every fileset in the domain can access the available storage. (See Chapter 3 for a discussion of quotas.)

2.3.4.1 Increasing Storage in Domains by Adding Volumes

A newly created domain consists of one volume, which can be a disk, disk partition, or logical volume. If you have AdvFS Utilities, you can increase

the size of the domain by adding one or more volumes. Figure 2–2 shows a graphic illustration of adding volumes to a domain. The volume you add initially contains no data. Once the volume is added, information will be stored on it as a result of adding files to the domain, as the `vfast` utility operates (see Section 5.8), if you migrate files to it (see Section 5.12), and if you balance the domain (see Section 5.11).

Figure 2–2: Enlarging a Domain



To add volumes, use the SysMan Menu utility called Manage an AdvFS Domain (see Appendix A), a graphical user interface (see Appendix E), 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
```

The `addvol` command supports shorthand names for LSM volume names. The following commands are equivalent:

```
# addvol dom1 testdg.vol1  
# addvol dom1 /dev/vol/testdg/vol1
```

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.

Performance might improve if, instead of increasing the size of a DVN3 domain, you upgrade to the new file structure (Section 2.3.3.4) before increasing its size.

You cannot add a volume to the root domain unless you are running a cluster. (See the *Cluster Administration* manual.) 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, see Section 6.4.9.

Caution

Do not use the `addvol` command to add a volume containing data that you want to keep. Executing the command destroys the data on that volume. If you have accidentally used the `addvol` command, you might recover some of your data by using the `/sbin/advfs/salvage` utility. (See Section 6.2.6.)

To add volumes to a domain, do the following:

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. Add the new volume to the domain.

If you add a partition and it overlaps with a partition that has an entry in the `/etc/fdmns` directory (that is, the partition is associated with a domain), you get an error message. You can override the message. Alternatively you can use the `addvol` command with the `-F` option to bypass this test. See `addvol(8)` for more information.

Caution

Use the `-F` option with extreme caution. Disabling the overlap check can result in extensive data loss. (See `addvol(8)`.)

3. If you are not running the `vfast` utility (see Section 5.8), it is a good idea to run the `balance` utility to even the file distribution between the volumes.

The following example adds one disk, the volume `/dev/disk/dsk3c` to `domain_1`:

```
# showfdmn domain_1
  Id          Date Created      LogPgs  Version  Domain Name
2bb0c594.00008570  Fri Mar 22 12:33 2002 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
```

2.3.4.2 Increasing Storage in Domains by Exchanging Volumes

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 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.

2.3.4.3 Increasing Storage in Domains by Extending an Existing Volume

You can increase the amount of storage available in your domain by enlarging the size of an existing volume. For example, LSM and hardware RAID controllers support dynamically increasing the size of a volume. Because expanding the size of an LSM or hardware RAID volume is done independently of AdvFS, you must notify the domain when the size of a volume changes by using the `mount` command with the `-o extend` option. You only need to mount one fileset with this option to increase the size of the domain.

If the fileset is not already mounted, enter:

```
mount -o extend domain#fileset /mount_point
```

If the fileset is already mounted, enter:

```
mount -u -o extend domain#fileset /mount_point
```

When the `-o extend` option is used, all additional storage is available to all filesets in the domain.

Note

Increasing the size of a domain by extending a volume is unidirectional. AdvFS does not support shrinking volumes.

2.3.5 Reducing the Size of Domains

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 off-loaded 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.

You can only use AdvFS commands to reduce the size of a domain. No analog to the mount command with the `-o extend` option exists to indicate to a domain that the storage has been reduced.

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 Menu utility called Manage an AdvFS Domain (see Appendix A), a graphical user interface (see Appendix E), 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
```

The `rmvol` command supports shorthand names for LSM volume names. The following commands are equivalent:

```
# rmvol dom1 testdg.voll  
# rmvol dom1 /dev/vol/testdg/voll
```

You can interrupt the `rmvol` process with `Ctrl/c` or by using the `kill -term` command without damaging your domain. (See `rmvol(8)`.) Files already

removed remain in their new location. Do not execute 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, execute the `chvol` command with the `-A` option 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. If you are not running the `vfast` utility, it is a good idea to 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 or that the `vfast` utility is 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 Apr 26 10:23 2002 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
```

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 5.13 for information on file striping.

2.3.6 Removing Domains

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.

You can use the `rmfdmn` command to remove a domain and inactivate filesets. This might present a security hole because the command changes no data on the volumes. You might be able to access your data by using the `/sbin/advfs/salvage` command. (See Section 6.2.6.) Instead, use the `rmfset` command to remove each fileset. (See Section 2.4.7.)

To remove a domain, use a graphical user interface (Appendix E), 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.

2.3.7 Renaming Domains

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 2.4.1.) To rename a domain, use a graphical user interface (Appendix E) or, from the command line do the following:

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_domain_name /etc/fdmns/new_domain_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 command line and 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
```

2.4 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.

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. For the advantages and disadvantages of different domain and fileset configuration, see Section 1.4.1.

Because filesets are managed independently of their physical storage, each fileset can be backed up independently (Chapter 4) and can be assigned its own quota limits (Chapter 3). 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.

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 must be unique within a domain. 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.

Filesets 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.

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. See Section 2.4.10 for a full explanation of fileset clones.

2.4.1 Designating Filesets in the `/etc/fstab` File

Add AdvFS filesets to the `/etc/fstab` file by listing them with an `advfs` file system 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 user and group quotas (see Section 3.2), include the `userquota` and `groupquota` options and the pass field numbers. No spaces can be left in the list of options delimited by commas; that is, from `rw` through `groupquota`.

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

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

```
acct_124#crdt /crdt advfs rw,userquota,groupquota 0 2
```

If you want only to add group quotas, for example, do not include the user quota designation. In this case the fileset entry would be:

```
acct_124#crdt /crdt advfs rw,,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 the `-a` (all) option that directs the command to operate only on file systems specified in this way.

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 files in the `userquota` and `groupquota` parameters.

For example, to relocate the `quota.user` file to the `d4` subdirectory in the `crdt` fileset and rename it `nq`, change the `/etc/fstab` entry to:

```
acct_124#crdt /crdt advfs rw,userquota=/d4/nq,groupquota 0 2
```

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

2.4.2 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 the following command:

```
showfsets domain_name
```

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

```
# showfsets zso_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 to obtain similar information. (See Appendix E.)

2.4.3 Creating Filesets

A domain must contain at least one mounted fileset to be active. (See Section 2.3.3.) 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 3.) You can control the creation of frags for files that waste more than 5% of their allocated storage. (See Section 5.2.) You can set the characteristics of the fileset at creation by using the `mkfset` command or change them later by using the `chfsets` command. (See Section 2.4.9.)

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

```
mkfset domain_name fileset_name
```

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

```
# mkfset advertising coupons
```

To mount a fileset, you must create a mount-point directory for it. (See Section 2.4.5.)

See also Section 2.3.3, which explains domains.

2.4.4 Upgrading Filesets

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 2.3.3.)

You cannot mount DVN4 filesets 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 6.4.6 for information on avoiding other incompatibilities.

2.4.5 Mounting Filesets

As with traditional UNIX file systems, AdvFS filesets must be mounted to access them. You must create a mount-point directory if one does not exist. Filesets to be mounted must be compatible with the operating system on which they were created. (See Section 6.4.6.)

To mount a fileset, use the SysMan Menu utility called General File System Utilities — Mount File Systems (see Appendix A), a graphical user interface (see Appendix E), or enter the `mount` command from the command line:

```
mount domain_name#fileset_name mount_point
```

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

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

To automatically mount your fileset at reboot, it must be entered in the `/etc/fstab` file, as described in Section 2.4.1.

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 6.3.2.)

If you attempt to mount a fileset with an incorrect number of volumes, the mount operation fails. See Section 6.3.3.2 and `advscan(8)` for more information.

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 6.4.6.)

You can specify temporary **atomic-write data logging** in your mount operation. (See Section 5.5.)

2.4.6 Unmounting Filesets

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 Menu utility called General File System Utilities — Dismount a File System (see Appendix A), a graphical user interface (see Appendix E), or enter the `umount` command from the command line:

```
umount mount_point
```

For example, to unmount the fileset `coupons` in the domain `advertising` that was mounted in Section 2.4.5:

```
# umount /coupons
```

2.4.7 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 before you can remove it. You must also remove its fileset clone. If you have set up a trashcan directory (Section 2.6) for the fileset, it is removed when you remove the fileset.

To remove a fileset, use the SysMan Menu utility called Manage an AdvFS Domain (see Appendix A), a graphical user interface (see Appendix E) 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 most secure way to remove all filesets in a domain is by using the `rmfset` command. The utility destroys pointers to the metadata for all the files in the fileset so you cannot recover them by using the `/sbin/advfs/salvage` utility. (See Section 6.2.6.) It is faster to delete filesets by using the `rmfdmn` command to remove the domain. However, if you do this, you might be able to access some of the data, because the `rmfdmn` command removes the definition of the domain in the `/etc/fdmns` directory and relabels the volumes, but it does not alter any data on the volume.

2.4.8 Renaming Filesets

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 Menu utility called Manage an AdvFS Domain (see Appendix A), a graphical user interface (see Appendix E) or, from the command line, enter the following command:

```
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 2.4.1.) If you do not do this, AdvFS does not mount the fileset when the system is booted.

For example, to rename the `sad` fileset in `dmn_1` that is mounted at `/mount_point` to `happy` using the command line and `vi` editor:

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

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

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

Edit this line as follows:

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

Mount the fileset:

```
# mount /mount_point
```

Note

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

2.4.9 Changing the Characteristics of Filesets

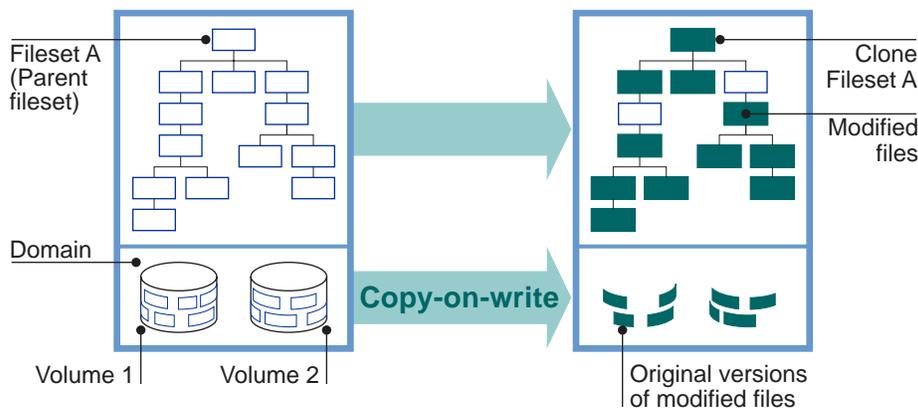
You can change fileset characteristics with the `chfsets` command. Change the fileset quota limits by using the `-F`, `-f`, `-B`, and `-b` options. (See Section 3.3.2.) Use the `-o` option to turn fragment files on and off (Section 5.2), to enable or disable object safety (Section 6.4.11), and to enable or disable the data management API (DMAPI) (Appendix D).

2.4.10 AdvFS Fileset Clones

If you have the optional AdvFS Utilities license, and you are root user, you can back up your files using an AdvFS fileset clone. A fileset clone is a read-only snapshot of fileset structure information (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. Therefore a copy of the system as it was at the time of creating the clone remains for the life of the clone. (This process is called **copy-on-write**.)

The only data in the fileset clone is a copy of data that has been modified, so the fileset clone is usually smaller than the original fileset. Figure 2–3 illustrates the relationship between a parent fileset and its clone.

Figure 2–3: Cloning a Fileset



Fileset clones increase the availability of data because they do the following:

- 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 it as a backup source for the `vdump` command. (See Chapter 4 for more information.)
- 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. Data caching is not applicable to clones.

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 fileset clone might grow very large. AdvFS cannot 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.

You cannot clone a DMAPi-enabled fileset.

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

2.4.10.1 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 Menu utility called Manage an AdvFS Domain (see Appendix A), a graphical user interface (see Appendix E), 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
```

See Section 4.3.11 for command-line instructions on using cloning as a source for backup. See Section E.4.3 for an explanation of cloning with the AdvFS GUI.

2.4.10.2 Mounting and Unmounting an AdvFS Fileset Clone

Clones are mounted and unmounted in the same manner as any other fileset. (See Section 2.4.5 and Section 2.4.6.)

2.4.10.3 Removing an AdvFS Fileset Clone

Clones are removed in the same manner as any other fileset. (See Section 2.4.7.)

2.4.10.4 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. This new clone is a snapshot of the fileset at a later point in time than the deleted clone.

2.5 Configuring an AdvFS root File System

Configuring AdvFS for the root file system has several advantages:

- You can start quickly after a crash. You do not have to run the `fsck` utility.
- One set of tools manages 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.
- You can use AdvFS with mirroring to protect the root file system. Your root file system remains viable even if there is a media failure.

The following are restrictions on the AdvFS root file system:

- Unless you are running a cluster (see the *Cluster Administration* manual), 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 name that you assign must be entered in the `/etc/fstab` file.

Structure the root domain to 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 operating system installation, or you can convert your existing root file system after installation. When you install AdvFS as the root file system during the initial installation, root defaults to the `a` 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 operating system installation.

2.5.1 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` command with the `-u` option:

```
# 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. See `mount(8)` for more information.

2.5.2 Changing the Name of the root Domain

You can change the name of the root domain as you would any other domain. (See Section 2.3.7.) 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. Both need to be updated if you change the name.

2.5.3 Changing the Name of the root Fileset

You can change the name of the root fileset as you would any other fileset. (See Section 2.4.8.) There are, however, two complications:

- The `renamefsset` 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. 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, do the following:

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 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 the `/etc/fstab` file in single-user mode using an editor to proceed.

2.6 Setting Up Trashcans for File Recovery

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 at the time 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 and the trashcan is deleted.
- Deleted files in an attached trashcan count against your quota.
- When you delete files in the trashcan directory, they are unrecoverable.

Table 2–1 lists and defines the commands for setting up and managing a trashcan.

Table 2–1: Trashcan Commands

Command	Description
<code>mktrashcan</code>	Creates the trashcan
<code>shtrashcan</code>	Shows the contents of the trashcan
<code>rmtrashcan</code>	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
```

Managing Quotas

You can 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; they only apply to non-root users.

This chapter discusses the following topics:

- Quotas and grace periods (Section 3.1)
- User and group quotas (Section 3.2)
- Fileset quotas (Section 3.3)
- Methods of verifying usage (Section 3.4)

3.1 Introducing Quotas

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

- AdvFS differentiates between quota maintenance and quota enforcement. Quota information is always maintained, but you can activate and deactivate enforcement.
- AdvFS supports fileset quotas; that is, you can set quota limits to control the size of 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. Quota files are kept in the root directory of the fileset by default.

- 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. (See Chapter 4 for a description of backup procedures.)

3.1.1 Quota Limits

Limits are set on disk usage (number of blocks) or on number of files (inodes) or both. Table 3–1 shows the size limits for both types of quota values.

Table 3–1: 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 two 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 2.3.3.4 for more information.)

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 (see Section 3.1.2) 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 has 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 do the following:

- 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 6.4.5 for directions if you need to write beyond your quota limits.

3.1.2 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. 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 old limit.

You can set grace periods for the following:

- Users

For each fileset you can set only one grace period for all users. (See Section 3.2.2.) 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 a group. (See Section 3.2.2.) 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 3.3.2.) Any user or group can cause the fileset to exceed its soft limit and thereby invoke 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. You can specify the grace period in days, hours, minutes, or seconds. This period can be changed. Setting the grace period to 0 days imposes the default grace period. Setting the grace period to 1 second turns off the grace period.

3.1.3 Summarizing Quota Characteristics

Table 3–2 compares user and group quotas with fileset quotas.

Table 3–2: Comparing User and Group Quotas with Fileset Quotas

	User and Group	Fileset
Quota files	In files named <code>quota.user</code> and <code>quota.group</code> (Section 3.2.1)	Part of fileset metadata; not visible
Setting quota limits	Change the mount options in the <code>/etc/fstab</code> file, mount the fileset, and run the <code>edquota</code> command (Section 3.2.2)	Run the <code>chfsets</code> command (Section 3.3.2)
Setting grace period	Use the <code>edquota</code> command with the <code>-ut</code> or <code>-gt</code> option (Section 3.2.2)	Same as group grace period (Section 3.3.2)
Enabling quotas	Run the <code>quotaon</code> command or, for quota enforcement at reboot, use the <code>rcmgr</code> command (Section 3.2.4 and Section 3.2.5)	Setting quota limits turns them on (Section 3.3.4)
Disabling quotas	Use the <code>quotaoff</code> command or remove <code>QUOTA_CONFIG="yes"</code> from the <code>/etc/rc.config.common</code> file (Section 3.2.6)	Use the <code>chfsets</code> command to set quota limits to 0 (Section 3.3.5)
Checking quota limits	Use the <code>edquota</code> command with <code>-u</code> or <code>-g</code> option	Use the <code>chfsets</code> or <code>showfsets</code> command

3.2 User and Group Quotas

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

3.2.1 User and Group 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 and cannot be deleted although quotas need not be enforced. 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 (Section 1.3.3); that is, there are holes in the file where no user IDs or group IDs fall. If you execute the `ls -l` command to view the space spanned by the `quota.user` or the `quota.group` file, you will not see the actual disk usage. For example, to view the `quota.user` file, enter:

```
# 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 1K byte blocks the file actually uses. For example, to view the `quota.user` file, enter:

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

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

3.2.2 Setting User and Group 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 is ignored 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. 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 3.3.2.2 for more information.)

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 (Section 2.4.1) and mount the fileset if it is not mounted.

2. Enter the `edquota` command with the appropriate options to set user or group quotas or the grace period. 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. Only filesets for which quotas can be set will be visible. If a fileset is missing, repeat step 1.
3. When the user or group quota information is displayed, change the numbers shown in the hard and soft limit 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. An ASCII file that can be edited is displayed.
5. When the grace period information is displayed, change the numbers shown for the grace period. Then, exit the editor, saving the changes.

For more information on setting quotas and grace periods, see `edquota(8)`.

The quota limits you set for a fileset do not become effective until you activate them by using the `quotaon` command or at system start-up. (See Section 3.2.4 and Section 3.2.5 for more information.) If you are changing quota limits and quotas are already enforced for a fileset, the new limits become effective immediately.

Once you have set quotas for a single user, you can use the `edquota` command with the `-p` option, which allows that user's entry to serve as a prototype to be applied to other users that you specify. (See Section 3.2.3 for instructions.) If you do not use the `-p` option and you enter the `edquota` command with more than one user name, an editor window is generated for each user specified.

3.2.2.1 Setting User Quotas

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

1. If they do not already exist, add quota mount-point options to filesets in the `/etc/fstab` file (Section 2.4.1) that you want to set quotas for and mount the filesets that are not mounted.

```
domain_1#geb1 /geb1 advfs rw,userquota,groupquota 0 2
domain_2#geb3 /geb3 advfs rw,userquota,groupquota 0 2
domain_4#geb4 /geb4 advfs rw,userquota,groupquota 0 2
```

2. Enter the `edquota` command with the `-u` option followed by the user name to display the quota-setting file. If a fileset is missing from the `edquota` output, repeat step 1.

```
# edquota -u user5
Quotas for user user5:
/geb1: blocks in use: 0, limits (soft = 0, hard = 0)
      inodes in use: 0, limits (soft = 0, hard = 0)
/geb3: blocks in use: 0, limits (soft = 0, hard = 0)
      inodes in use: 0, limits (soft = 0, hard = 0)
/geb4: 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 1K byte blocks and the number of files currently in use for each fileset. You cannot change them. In this example, the filesets are newly created. Soft and hard limits of 0 (zero) indicate that no limits have been set.

3. Change the numbers shown in the `soft` and `hard` fields to set limits for the user you specified. If you specified more than one user name, the values affect all users named.

```
/geb1: blocks in use: 0, limits (soft = 0, hard = 0)
      inodes in use: 0, limits (soft = 0, hard = 0)
/geb3: blocks in use: 0, limits (soft=5000, hard=10000)
      inodes in use: 0 limits (soft= 100, hard= 200)
/geb4: blocks in use: 0, limits (soft = 0, hard = 0)
      inodes in use: 0, limits (soft = 0, hard = 0)
```

4. Exit the editor, saving the changes.

If quotas are already activated for fileset `geb3`, the new limits become effective immediately. If quotas are not yet enforced for the fileset, the limits become effective as soon as quotas are enforced. (See Section 3.2.5 for a discussion of activating quotas manually.)

If both user and group quotas are set, the lower of the two is enforced.

3.2.2.2 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. To turn off the grace period, set it to 1 second. 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 the user grace period, the user is limited by the group grace period.

The following procedure sets the grace period for all users of fileset `geb3`:

1. If they do not already exist, add quota mount-point options to filesets in the `/etc/fstab` file (Section 2.4.1) that you want to set quotas for and mount the filesets that are not mounted.

```
domain_1#geb1 /geb1 advfs rw,userquota,groupquota 0 2
domain_2#geb3 /geb3 advfs rw,userquota,groupquota 0 2
domain_4#geb4 /geb4 advfs rw,userquota,groupquota 0 2
```

2. Enter the `edquota` command with the `-ut` option to display the grace period-setting file. If a fileset is missing from the `edquota` output, repeat step 1.

```
# edquota -ut
Time units may be: days, hours, minutes, or seconds
Grace period before enforcing soft limits for users:
/geb1: block grace period: 7 days, file grace period: 7 days
/geb3: block grace period: 7 days, file grace period: 7 days
/geb4: block grace period: 7 days, file grace period: 7 days
```

3. Change the numbers shown in the block grace period and file grace period fields to set the user grace period for the `geb3` fileset.

```
Time units may be: days, hours, minutes, or seconds
Grace period before enforcing soft limits for users:
/geb1: block grace period: 7 days, file grace period:7 days
/geb3: block grace period: 2 days, file grace period:3 days
/geb4: block grace period: 7 days, file grace period:7 days
```

4. Exit the editor, saving the changes.

If quotas are already activated for the fileset `geb3`, the new grace period becomes effective immediately. If a user has already exceeded the soft limit, the new grace period becomes effective when the usage drops below the soft limit. If quotas are not yet enforced for the fileset, the grace period becomes effective as soon as quotas are activated. (See Section 3.2.5 for a discussion of activating quotas manually.)

3.2.2.3 Setting Group Quotas

The following procedure sets quotas on `geb3` for the group `rsgusers`:

1. If they do not already exist, add quota mount-point options to filesets in the `/etc/fstab` file (Section 2.4.1) that you want to set quotas for and mount the filesets that are not mounted.

```
domain_1#geb1 /geb1 advfs rw,userquota,groupquota 0 2
domain_2#geb3 /geb3 advfs rw,userquota,groupquota 0 2
domain_4#geb4 /geb4 advfs rw,userquota,groupquota 0 2
```

2. Enter the `edquota` command with the `-g` option. If you specify more than one group name, the edits affect all groups named. If a fileset is missing from the `edquota` output, repeat step 1.

```
# edquota -g rsgusers
Quotas for group rsgusers:
/geb1: blocks in use: 0, limits (soft=0, hard=0)
      inodes in use: 0, limits (soft=0, hard=0)
/geb3: blocks in use: 0, limits (soft=0, hard=0)
      inodes in use: 0, limits (soft=0, hard=0)
/geb4: 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 1K byte 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. Change the numbers shown in the `soft` and `hard` fields to set limits for the group you specified.

```
/geb1: blocks in use: 0, limits (soft=0, hard=0)
      inodes in use: 0, limits (soft=0, hard=0)
/geb3: blocks in use:0, limits(soft=60000, hard=80000)
      inodes in use:0, limits(soft= 6000, hard= 8000)
/geb4: blocks in use: 0, limits (soft=0, hard=0)
      inodes in use: 0, limits (soft=0, hard=0)
```

4. Exit the editor, saving the changes.

If quotas are already activated for fileset `geb3`, these limits become effective immediately. If quotas are not yet enforced for `geb3`, these limits become effective as soon as quotas are activated. (See Section 3.2.5 for a discussion of activating quotas manually.) If both user and group quotas are set, the lower of the two is enforced.

3.2.2.4 Setting the Group Grace Period

When you impose soft limits, you can set one grace period per fileset for all groups. The group grace period is the same as the fileset grace period. (See Section 3.3.2.2.) If you do not specify a grace period, the grace period remains the AdvFS default of 7 days. To turn off the grace period, set it to 1 second. 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 `geb3`:

1. If they do not already exist, add quota mount-point options to filesets in the `/etc/fstab` file (Section 2.4.1) that you want to set quotas for and mount the filesets that are not mounted.

```
domain_1#geb1  /geb1  advfs rw,userquota,groupquota 0 2
domain_2#geb3  /geb3  advfs rw,userquota,groupquota 0 2
domain_4#geb4  /geb4  advfs rw,userquota,groupquota 0 2
```

2. Enter the `edquota` command with the `-gt` option to display the grace period-setting file. If a fileset is missing from the `edquota` output, repeat step 1.

```
# edquota -gt
Time units may be: days, hours, minutes, or seconds
Grace period before enforcing soft limits for groups:
/geb1: block grace period: 7 days, file grace period:7 days
/geb3: block grace period: 7 days, file grace period:7 days
/geb4: block grace period: 7 days, file grace period:7 days
```

3. Change the numbers shown in the `block grace period` and `file grace period` fields to set the group grace period for the `geb3` fileset.

```
Time units may be: days, hours, minutes, or seconds
Grace period before enforcing soft limits for groups:
/geb1: block grace period: 7 days, file grace period:7 days
/geb3: block grace period:12hours, file grace period:5 days
/geb4: block grace period: 7 days, file grace period:7 days
```

4. 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 `geb3`. 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 enforced, the group grace period becomes effective as soon as they are activated. (See Section 3.2.5 for a discussion of activating quotas manually.)

3.2.2.5 Setting a Temporary Grace Period

You can limit the life span of files in a fileset. If you set a temporary grace period, users can allocate files without limit during the time specified. After that time, no more files can be allocated until all files that were created during the grace period are removed. Setting a temporary grace period might be useful for situations where files are needed for a short time.

To set a temporary grace period, enter the `edquota` command with the `-t` option and set the soft limit for the number of blocks or inodes to 1 and the hard limit for the number of blocks or inodes to 0.

3.2.3 Setting User and Group Quotas for Multiple Users and Groups

You can modify quotas for a list of users or groups without accessing and entering values for each one individually. 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.

To set up all users to have the same quota, do the following:

1. Use the `edquota` command with the `-u` or the `-g` option to establish quotas for a single user or for a group.
2. Use the `edquota` command with the `-p` option to apply the quotas set up for the first user to the others.

See `edquota(8)` for more information.

3.2.3.1 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` as described in Section 3.2.2.1.

```
# edquota -u user5
Quotas for user user5:
/geb1:blocks in use:0, limits(soft= 0,hard= 0)
      inodes in use:0, limits(soft= 0,hard= 0)
/geb3:blocks in use:1 ,limits(soft= 5000,hard= 10000)
      inodes in use:4, limits(soft= 100,hard= 200)
/geb4:blocks in use:2, limits(soft= 0,hard= 0)
      inodes in use:1, limits(soft= 0,hard= 0)
```

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:
/geb1:blocks in use:0, limits(soft= 0,hard= 0)
      inodes in use:0, limits(soft= 0,hard= 0)
/geb3:blocks in use:1 ,limits(soft= 5000,hard= 10000)
      inodes in use:4, limits(soft= 100,hard= 200)
/geb4:blocks in use:2, limits(soft= 0,hard= 0)
      inodes in use:1, limits(soft= 0,hard= 0)
```

3.2.3.2 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` as described in Section 3.2.2.3:

```
# edquota -g rsgusers
Quotas for group rsgusers:
/geb1:blocks in use:0, limits(soft=    0,hard=    0)
      inodes in use:0, limits(soft=    0,hard=    0)
/geb3:blocks in use:0, limits(soft= 60000,hard= 80000)
      inodes in use:0, limits(soft=  6000,hard=  8000)
/geb4:blocks in use:0, limits(soft=    0,hard=    0)
      inodes in use:0, limits(soft=    0,hard=    0)
```

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:
/geb1:blocks in use:0, limits(soft=    0,hard=    0)
      inodes in use:0, limits(soft=    0,hard=    0)
/geb3:blocks in use:0, limits(soft= 60000,hard= 80000)
      inodes in use:0, limits(soft=  6000,hard=  8000)
/geb4:blocks in use:0, limits(soft=    0,hard=    0)
      inodes in use:0, limits(soft=    0,hard=    0)
```

3.2.4 Activating User and Group Quotas at System Start-Up

Follow these steps to start user and group quota enforcement automatically during system initialization:

1. Edit the `/etc/fstab` file entry to add quota file mount-point options. See Section 2.4.1 for more information about mount-point options.
2. Use the `rcmgr` command to add the `QUOTA_CONFIG` option to the `/etc/rc.config.common` file.

```
/usr/sbin/rcmgr -c set 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` file.

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 start-up for AdvFS file systems with quota options in the `/etc/fstab` file, execute 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 ""
```

3.2.5 Activating User and Group Quotas Manually

After you have added `userquota` and `groupquota` entries to the `/etc/fstab` file for your mounted filesets and have run the `edquota` command to set limits and grace period, run the `quotaon` command to activate quotas.

If your system is set up to initialize quotas at system start-up (Section 3.2.4), 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 (Section 3.2.6), 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
/geb1:  group quotas turned on
/geb1:  user quotas turned on
/geb3:  group quotas turned on
/geb3:  user quotas turned on
/geb4:  group quotas turned on
/geb4:  user quotas turned on
```

By default, both user and group quotas are affected by the `quotaon` command. 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 `quotaon(8)` for more information.

3.2.6 Deactivating User and Group 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. You can choose to deactivate 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 `quotaoff(8)` for more information.

The `umount` command turns off quotas before it unmounts a fileset. You cannot turn on quotas for an unmounted 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 off quotas for a user or group, do the following:

1. Run the `quotaoff` command to prevent quotas from being enforced for the fileset.
2. Enter the `edquota` command and set quota limits to 0 (zero).
3. Remove the `userquota` and `groupquota` entries for the fileset in the `/etc/fstab` file.

3.3 Fileset Quotas

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

3.3.1 Fileset 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.

3.3.2 Setting Fileset 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 enforced whenever you mount the fileset.

Fileset quotas have a grace period. (See Section 3.3.2.2 for more information.) If you do not set a grace period, the grace period remains at the AdvFS default grace period of 7 days. To turn off the grace period, set it to 1 second.

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

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

The command displays both the old and new limits.

3.3.2.1 Setting Fileset Quotas

The following procedure sets fileset quotas for the `student_files` fileset in the `year1` domain. Unlike other 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, execute the `showfsets` command.

```
# showfsets -k year1 student_files
student_files
  Id          : 2feff762.00034e3f.1.8001
  Clone is    : stufiles_clone
  Files       :          7,  SLim=    0,  HLim=    0
  Blocks (1k) :         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.

```
# chfsets -F 1000 -f 2000 -B 25000 -b 50000
year1 student_files
```

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

```
# showfsets -k year1 student_files
student_files
  Id          : 2feff762.00034e3f.1.8001
  Clone is    : stufiles_clone
  Files       :      7,  SLim= 1000,  HLim=  2000
  Blocks (1k) :    118,  SLim= 25000,  HLim= 50000
  Quota Status : user=on  group=on
  Object Safety: off
  Fragging    : on
  DMAPi       : off
```

3.3.2.2 Setting the Grace Period

The fileset grace period is the same as the group grace period and cannot be modified independently. Therefore, if you use the `edquota` command with the `-gt` option to change the grace period for which a group can exceed its soft limits, you also change this grace period for the fileset. See Section 3.2.2.4 for an explanation of setting the group (and fileset) grace period.

3.3.3 Setting Fileset Quotas for Multiple Filesets

You can set quota limits for more than one fileset in a domain by listing more than one fileset name when you run the `chfsets` command.

For example, to change the hard limits for the `data` and `data2` filesets in `domain_2`, enter the names of both filesets after the `chfsets` command:

```
# chfsets -b 1000 -f 200 domain_2 data data2
```

3.3.4 Activating Fileset Quotas

Running the `chfsets` command immediately activates fileset quotas. No further steps are needed. Fileset quotas are in effect whenever you mount the fileset.

3.3.5 Deactivating Fileset Quotas

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

3.4 Verifying Fileset and Disk Space Usage

You can monitor file and disk space usage by using commands such as `df`, `showfdmn`, `/sbin/advfs/vdf`, and `showfsets`. For general information on displaying disk usage or if your disk usage information appears to be corrupt, see Section 6.4.1. Commands that display the address range of a file, such as `ls`, make a quota file, which is sparse, look larger than it actually is.

3.4.1 Examining User and Group Quotas

If you are enforcing user and group quotas, you can periodically verify your quota setup. 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 1K byte blocks.

The commands shown in Table 3–3 are useful for examining disk space and file usage for filesets for which user and group quotas are enforced.

Table 3–3: User and Group Usage Commands

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

3.4.1.1 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
```

3.4.1.2 Summarizing Fileset Ownership

The `quot` command displays block usage and the number of files in the mounted filesets 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 for each user.

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

The `quot` command with the `-f` option displays both the size and number of files for each user.

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

3.4.1.3 Displaying Disk Usage and Limits

The `quota` command displays the block usage, the number of files, and the quotas for a user or group for mounted filesets that have `userquota` and `groupquota` entries in the `/etc/fstab` file. 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 enforced.

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 indicates that a soft limit has been exceeded. The grace period is not displayed unless the soft limit has been exceeded. Execute the `edquota` command to view existing quotas.

The following example displays quota information for the user `runner1`:

```
# quota -u runner1
Disk quotas for user runner1 (uid 446):
Filesystem  blocks  quota  limit  grace  files  quota  limit  grace
/           60     100    150
/usr        5071*  5000   10000  24:40   2     20    40
/mile1      816    20000  30000
/mile2     22032  50000  200000
/mile3     2344   10000  15000   370    1000  2000
/mile4     18023* 10000  20000  7days   3     100   150
/mile5     32012* 20000  50000  7days   0     2000  3000
```

The following example displays quota information for the group `kingco`:

```
# quota -g kingco
Disk quotas for group kingco (gid 15):
Filesystem  blocks  quota  limit  grace  files  quota  limit  grace
/           118    200    300
/usr       13184* 10000  20000  7days   2     40    80
/mile1     36136 100000 200000
/mile2     44064 200000 400000
           4     2000  4000
```

/mile3	3587	30000	60000	628	3000	5000
/mile4	51071	150000	300000	6	1050	1800
/mile5	61044	100000	200000	3	10000	20000

3.4.1.4 Verifying User and Group 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`, which has no inconsistencies.

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

The following example checks all filesets that have quotas defined in the `/etc/fstab` file. In this display, inodes is the number of files and blocks is the block usage. Here 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 (/set_1)
/usr:root    fixed group quota: inodes   3057 -> 3022
                                   blocks 100616 -> 123440
/usr:system  fixed group quota: inodes   2483 -> 2488
                                   blocks 91721 -> 114568
/usr:adm     fixed group quota: inodes   280 -> 240
                                   blocks  487 ->  464
```

3.4.1.5 Summarizing User and Group Quotas by Fileset

The `repquota` command displays the actual disk usage and quotas for the specified filesets. The fileset must be mounted and 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 lists 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 `/set_1`:

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

The following example displays user and group quota information for all mounted filesets that have quotas defined in the `/etc/fstab` file. 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      0   2488    0    0
daemon  --   144    0    0      0     1    0    0
uucp    --   801    0    0      0     8    0    0
mem     --  1096    0    0      0    10    0    0
bin     -- 108989    0    0      0   3219    0    0
mail    --   209    0    0      0     2    0    0
terminal --    56    0    0      0     2    0    0
adm     --   464    0    0      0    240    0    0
operator --   392    0    0      0     3    0    0
211     --  6937    0    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      0   3022    0    0
bi       -- 102534    0    0      0   2940    0    0
uucp    --   729    0    0      0     7    0    0
adm     --     1    0    0      0     1    0    0
user5   --    15   18   24      0     1    0    0
kraetsch --  6937    0    0      0    35    0    0
*** Report for group quotas on /set_1 (domain_1#set_1)
      Block limits          File limits
Group      used  soft  hard  grace  used  soft  hard  grace
system  --  22816    0    0      0     50    0    0
daemon  --  12088    0    0      0     82    0    0
*** Report for user quotas on /set_1 (domain_1#set_1)
      Block limits          File limits
User      used  soft  hard  grace  used  soft  hard  grace
root     --  34088    0    0      0    123    0    0
user5   --   816 20000 30000    9    350   500
```

```

*** Report for group quotas on /set_3 (domain_2#set_3)
      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 /set_3 (domain_2#set_3)
      Block limits          File limits
User       used  soft  hard  grace  used  soft  hard  grace
root      --  1593    0    0          6    0    0

```

3.4.2 Examining Fileset Quotas

Fileset quota information is accessed with the general commands used to examine filesets. The commands shown in Table 3–4 are useful for this task.

Table 3–4: User and Group Usage Commands

Command	Description
df	Displays disk usage by fileset
showfsets	Displays fileset quota limits when the <code>-q</code> option is specified
showfdmn	Displays domain information
/sbin/advfs/vdf	Reformats output from other fileset usage commands

3.4.2.1 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.
- If 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.

Output of the `df` command does not include space used by metadata or quota files.

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

```
# df /fileset_1
Filesystem      512-blocks  Used Avail  Capacity  Mounted on
domain_1#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).

In the following example, the filesets `jan` and `mar` each can use all of the available disk space from the volumes in the `credit` domain:

```
# df
Filesystem      512-blocks  Used      Avail Capacity  Mounted on
credit#jan      2000000    390820    98864      80%  /jan
credit#mar      2000000    271580    98864      73%  /mar
```

3.4.2.2 Displaying Fileset Limits

Use the `showfsets` command with the `-q` option to display file usage, hard and soft limits, and grace period for the filesets in the specified domain. The command displays 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 `domain_1`:

```
# showfsets -q domain_1
          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 indicates that the soft limit for block usage is exceeded. An asterisk (*) would indicate that the hard limit has been reached. The limit for number of files has not been reached.

3.4.2.3 Displaying Domain Space

Use the `showfdmn` command to obtain 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 including 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 Sun Jan 20 17:01 2002   512      4 usr_domain

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

3.4.2.4 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.

It is a good idea to run the `vdf` command on a system that is not active. Because the command executes a sequence of utilities, a system that is actively adding, removing, and modifying files can cause `vdf` to generate inconsistent output.

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 `jba` that contains two filesets. Each has a quota limit of 60,000 blocks.

```
# /sbin/advfs/vdf -l jba
```

Domain	512-blocks	Metadata	Used	Available	Capacity
jba	266240	5824	29128	231288	13%

Fileset	QuotaLimit	Used	Available	Capacity
jbfsa	60000	20800	39200	35%
jbfs2a	60000	8328	51672	14%

The following example displays the domain `jbb` 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 jbb
```

Domain	512-blocks	Metadata	Used	Available	Capacity
jbb	266240	5824	29128	231288	13%

Fileset	QuotaLimit	Used	Available	Capacity
jbfsb	-	20800	231288	35%
jbfs2b	-	8328	231288	14%

Backing Up and Restoring Data

Backing up file-system data is a critical component of an overall strategy of preparing for application failure, system failure, and disaster scenarios. Designing a good backup strategy ensures that the data in the backup is usable when it is needed.

This chapter covers items to consider when planning your backup strategy:

- Assuring data and metadata integrity (Section 4.1)
- Creating a backup source (Section 4.2)
- Methods and tools that can be used to perform the backup (Section 4.3)

4.1 Data and Metadata Integrity

Before backing up your data, ensure that the source of your backup, both the data in your files and the AdvFS metadata (file structure information), are in a consistent state and will not change during the time that the backup is running.

4.1.1 Ensuring Data Integrity

A backup is only useful if the data it collects is consistent. Before beginning a backup, you must place your data files into such a state. This process has two steps.

1. Stop or quiesce your applications so that the current state of the data files is consistent from the application's standpoint.
2. Make sure that updates to the application's data files are written to permanent storage and are not cached either by the application or by the file system.

Failure to take these steps could result in a backup that is not usable because the data in it is inconsistent from an application standpoint.

Some applications, such as Oracle, provide explicit ways to quiesce and flush data to disk. Other applications do not have this functionality but use synchronous writing methods to ensure that if the application is stopped, the data written is on disk. Finally, some applications do not provide quiesce-and-flush functionality and do not use synchronous writing techniques.

4.1.1.1 Ensuring Data Integrity Using An Application's Quiesce Function

Some applications can quiesce and flush data to disk in preparation for backup. In Oracle, for example, this is known as Hot Backup Mode. To use it, enter the following Oracle SQL command:

```
ALTER TABLESPACE name BEGIN BACKUP
```

Then, once the source of the backup has been created (Section 4.2), enter the following Oracle SQL command to take the tablespace out of Hot Backup Mode:

```
ALTER TABLESPACE name END BACKUP
```

4.1.1.2 Ensuring Data Integrity Using Synchronous Writing

AdvFS provides ways to ensure that data written by an application is in permanent storage and is not in the application or file-system buffer cache:

- Use an `fsync()` system call to flush the cached data.
- Open a file with the `O_SYNC` or `O_DSYNC` flag to cause all cached data to be synchronously written to storage.

If an application using either of these techniques is cleanly shut down, you are guaranteed that all cached data has been written to permanent storage.

4.1.1.3 Ensuring Data Integrity Without Using Synchronous Writing

If an application does not have a quiesce function and does not call for synchronous writes, you can ensure that all data written by the application to its data files has been written to permanent storage before you begin your backup by doing one of the following:

- Mounting the fileset with the `-o sync` option. This forces updates to files in the fileset to be synchronously written to storage. As using this option can cause a significant decrease in performance to all the applications using that fileset, it is preferable to have applications use one of the other methods described to flush only the updates that need to be flushed to storage.
- Entering the `chfile` with the `-l on` option. This causes updates to a file to be synchronously flushed to storage regardless of the I/O mode requested by the applications using the file. See Section 5.5.1 for a discussion of I/O mode.
- Cleanly stopping the application. This causes the operating system's `smoothsync` daemon to flush all cached data to disk. After the application has been stopped, wait at least twice as many seconds as the value of `smoothsync_age` kernel tunable before starting your backup or

creating the backup source. The value of the `smoothsync_age` tunable can be retrieved by the command:

```
# sysconfig -q vfs smoothsync_age
```

4.1.2 Ensuring Metadata Integrity

After you have ensured that your data files are consistent from an application standpoint, you must take steps to ensure that the AdvFS metadata is also consistent. Failure to do so could result in AdvFS domain panics, system panics, or user data corruption. To guarantee consistent AdvFS metadata in your backup source, do one of the following:

- Freeze the domain using the `freezefs` command (Section 4.1.2.1)
- Unmount all filesets in the domain (Section 4.1.2.2)

4.1.2.1 Ensuring Metadata Integrity by Freezing Domains

The `freezefs` command causes a domain to enter into a metadata-consistent state and guarantees that it stays that way until the specified freeze time expires or it is explicitly thawed with the `thawfs` command. All of the filesets in the domain are frozen. All metadata, which could be spread across multiple volumes or logical units (LUNs), is flushed to disk and does not change for the duration of the freeze.

When you freeze a file system, all in-process file-system operations are allowed to complete. Some file-system operations that do not require metadata updates (for example, read) work normally even if the file system is frozen.

Once frozen, the file system remains in this metadata-consistent state until it is thawed:

- By timing out
- By the issuance of the `thawfs` command
- In a cluster, by shutting down any node of a frozen file system or if any cluster member fails

By default, the `freezefs` command freezes a file system for 60 seconds. Use the `-t` option to specify a lesser or greater time-out value in seconds or to specify that the domain remains frozen until being thawed by the `thawfs` command. See `freezefs(8)` and the *Cluster Administration* manual for more information.

4.1.2.2 Ensuring Metadata Integrity by Unmounting Filesets

You can ensure that AdvFS metadata doesn't change in a domain by unmounting all of the filesets in the domain. Do not run any AdvFS utilities

that can run on unmounted domains when you want to ensure metadata integrity.

4.2 Creating a Backup Source

Once you have created your backup source from the mounted fileset, you can complete the backup. To create the source, use one of the following:

- Original fileset (Section 4.2.1)
- Fileset clone (Section 4.2.2)
- Logical Storage Manager (LSM) split mirror (Section 4.2.3)
- Controller-based clone (Section 4.2.4)
- Controller-based snapshot (Section 4.2.5)

4.2.1 Using a Mounted Original Fileset as a Backup Source

In some situations you can use your original mounted fileset as the source for your backup.

4.2.1.1 Advantages and Disadvantages

Table 4–1 lists the advantages and disadvantages of mounting the original fileset as a backup source.

Table 4–1: Using a Mounted Original Fileset as a Backup Source

Advantages	Disadvantages
Simplicity. No new mount point needed.	During backup, files under mount point cannot be updated without risking data inconsistency in backup. Backup processing cannot be off-loaded to another host.

4.2.1.2 Ensuring Data and Metadata Integrity

To ensure that the data in the fileset is consistent, follow any procedure detailed in Section 4.1.

For backup applications that access the file system through the POSIX interface (such as the `vdump` utility), no special steps are needed to ensure the integrity of the metadata .

4.2.1.3 Preparing the Backup Source

Beyond ensuring data and metadata integrity, no special steps are needed to prepare a mounted original fileset for use as a backup source.

4.2.2 Using a Mounted Fileset Clone as a Backup Source

A fileset clone (Section 2.4.10) is a read-only snapshot of the data in an existing fileset. A fileset clone does not contain all the data at the time it is created. When you modify the data in your original files, AdvFS saves the data that existed in the original, page by page, into the fileset clone.

You must have the optional AdvFS Utilities to create a fileset clone. You must be the root user to clone a fileset. Only one clone can exist per fileset.

4.2.2.1 Advantages and Disadvantages

Table 4–2 lists the advantages and disadvantages of using a mounted fileset clone as a backup source.

Table 4–2: Using a Mounted Fileset Clone as a Backup Source

Advantages	Disadvantages
No additional storage hardware required. Domain must have space for clone.	Fileset clones created in same domain as fileset. Storage failures affect both original fileset and clone.
Simple to use. ^a	Cannot off-load processing of backup to another machine. Fileset clone must be mounted on same host as original fileset.
Off-line time for the original fileset limited to the time it takes to create a fileset clone, typically a few seconds.	Writing to clone impacts fileset I/O. Only one fileset clone can be created per original fileset at a time. If data spans more than one fileset, may need to stop application activity during cloning to ensure consistency.

^a You can create a fileset clone with the command line, the SysMan Menu utility called Manage an AdvFS Domain (see Appendix A) or with the AdvFS GUI (see Appendix E). You can also use the AdvFS API to write programs to fileset clones. See Appendix F.

4.2.2.2 Ensuring Data and Metadata Integrity

To ensure that the data in the fileset is consistent, follow any procedure detailed in Section 4.1. If the data spans more than one fileset, be certain that all associated filesets are consistent.

Because the fileset cloning process is done entirely within AdvFS, no special steps are needed to ensure the integrity of the AdvFS metadata.

4.2.2.3 Preparing the Backup Source

To create a fileset clone, ensure that the data in the original fileset is in a consistent state. Then create the fileset clone with the `clonefset` command.

The following example clones the `pssm` fileset in the `domain1` domain. The name of the new fileset clone is `pssm_clone`.

1. Create the clone and mount it.

```
# clonefs domain1 pssm pssm_clone
# mkdir /pssm_clone
# mount -t advfs domain1#pssm_clone /pssm_clone
```

2. Run the backup process using the `pssm_clone`. See Section 4.3.11 for more information.
3. Remove the clone.

```
# umount /pssm_clone
# rmfs domain1 pssm_clone
```

4.2.3 Using a Mounted Fileset from an LSM Split Mirror as a Backup Source

If the data you want to back up is stored in a domain built entirely on LSM mirrored volumes, you can split off an LSM mirror (detach a mirror), mount it, and use it as your backup source. You must be the root user to perform this operation. See the *Logical Storage Manager* manual for directions on how to detach a mirror.

4.2.3.1 Advantages and Disadvantages

Table 4–3 lists the advantages and disadvantages of using an LSM split mirror as a backup source.

Table 4–3: Using an LSM Split Mirror as a Backup Source

Advantages	Disadvantages
Full physical copy of data created. If storage hardware hosting original fileset fails, copy on split mirror not affected.	Requires LSM license.
Off-line time for the original fileset limited to the time it takes to split the LSM mirror, typically a few seconds.	Requires enough physical storage to hold at least two copies of data.
If data spans multiple filesets within a single domain, LSM split mirror serves as backup source for all of them.	More complexity because both LSM and AdvFS commands needed.
More than one LSM split mirror can be created for the same domain simultaneously.	Cannot off-load processing of backup to another machine. LSM split mirror fileset must be mounted on same host as original fileset.

4.2.3.2 Ensuring Data and Metadata Integrity

To ensure that the data in the fileset is consistent, follow any procedure detailed in Section 4.1.

If the storage making up the domain consists of only one LSM volume, no special steps are needed to ensure the integrity of the AdvFS metadata. If the storage spans multiple LSM volumes, see Section 4.1.2.

4.2.3.3 Preparing the Backup Source

The freeze and thaw steps are only necessary if your domain spans more than one LSM volume. If you are not sure, perform the commands anyway. They do not harm the domain. Freezing and thawing a single fileset freezes and thaws the entire domain.

The following example assumes that the original domain was comprised of two LSM volumes and that a mirror was detached from each. The data is in a domain that has a fileset named `data` mounted on `/data`. The two mirror volumes are called `/dev/vol/mirrorvol1` and `/dev/vol/mirrorvol2`.

1. Freeze the domain.

```
# /usr/sbin/freezefs /data
```

2. Execute the LSM commands necessary to detach the mirrors and reconstitute them as new LSM volumes. See the *Logical Storage Manager* manual for directions.
3. Once the LSM split mirror operations have completed, thaw the domain's metadata.

```
# /usr/sbin/thawfs /data
```

Now your applications can continue to run on the original domain.

4. Create a new domain directory using the two newly created LSM volumes.

```
# mkdir /etc/fdmns/mirror_domain
# ln -fs /dev/vol/mirrorvol1 /etc/fdmns/mirror_domain
# ln -fs /dev/vol/mirrorvol2 /etc/fdmns/mirror_domain
```

Do not use the `mkfdmn` command to create the new directory. Doing so causes the mirror to become inaccessible.

5. Create a mount-point directory and mount the `data` fileset created from the split mirror in `mirror_domain`.

```
# mkdir /backup
# mount -o dual mirror_domain#data /backup
```

The backup can now be performed from the `/backup` directory.

4.2.4 Using a Mounted Fileset from a Controller-Based Clone as a Backup Source

Some storage controllers, such as the Compaq HSG80 and HSV110, allow you to split off a hardware mirror using controller-based commands. Conceptually, this is similar to a split : mirror: there are two (or more) complete physical copies of the data and one is separated for backup.

4.2.4.1 Advantages and Disadvantages

Table 4–4 lists the advantages and disadvantages of using a controller-based clone as a backup source.

Table 4–4: Using a Mounted Fileset from a Controller-Based Clone as a Backup Source

Advantages	Disadvantages
Full physical copy of data is created. If storage hardware hosting original fileset fails, copy not affected.	Requires enough physical storage to hold at least two copies of data.
Off-line time for the original fileset limited to time it takes to perform controller-based cloning, typically a few seconds.	Complex because both controller and AdvFS commands needed.
Can off-load processing of backup to another machine. Controller-based clone can, in many cases, be mounted on a different host.	
Can serve as single backup source for all filesets in a domain.	
More than one controller-based clone can be created simultaneously for the same domain.	

4.2.4.2 Ensuring Data and Metadata Integrity

To ensure that the data in the fileset is consistent, follow any procedure detailed in Section 4.1.

If the storage making up the domain consists of only one hardware LUN, no special steps are needed to ensure the integrity of the AdvFS metadata. If the storage spans multiple hardware LUNs, see Section 4.1.2.

4.2.4.3 Preparing the Backup Source

The freeze and thaw steps are only necessary if your domain spans more than one LUN. If you are not sure, perform the commands anyway. They

do not harm the domain. Freezing and thawing a single fileset freezes and thaws the entire domain.

The following example assumes that the data is in a domain that has a fileset named `data` mounted on `/data`. The clones created are `/dev/disk/dsk25c` and `/dev/disk/dsk26c`.

1. Freeze the domain.

```
# /usr/sbin/freezefs /data
```

2. Execute the commands necessary to create the controller-based clone. For examples of these commands using the Compaq HSG80 controller, see the Best Practice *Using StorageWorks HSG80 Controller-Based Cloning and Snapshotting*.

3. Thaw the metadata for the domain.

```
# /usr/sbin/thawfs /data
```

Now your applications can continue to run on the original domain.

4. Create block device special files for the newly cloned LUNs. This should be done on the host that will mount the new backup source. The `/sbin/hwmgrr` command is asynchronous, so it could complete before the block device special files are actually created.

On a stand-alone system, enter:

```
# /sbin/hwmgrr -scan scsi
```

On a cluster, enter:

```
# /sbin/hwmgrr scan component -category scsi_bus -cluster
```

Every time a new LUN is created, a new set of device special files is created.

5. Identify the block device special files.

```
# /sbin/hwmgrr -view devices
```

6. Create the directory and symbolic links necessary to access the new domain on `/dev/disk/dsk25c` and `/dev/disk/dsk26c`.

```
# mkdir /etc/fdmns/clone_domain
# ln -fs /dev/disk/dsk25c /etc/fdmns/clone_domain
# ln -fs /dev/disk/dsk26c /etc/fdmns/clone_domain
```

Do not use the `mkfdmn` command to create the new domain. Doing so causes the cloned data to become inaccessible.

7. Create a directory and mount the data fileset in `clone_domain`.

```
# mkdir /backup
```

If the clone domain and fileset are not mounted on the original host:

```
# mount clone_domain#data /backup
```

If the clone domain and fileset are mounted on the original host:

```
# mount -o dual clone_domain#data /backup
```

The backup can now be performed from the /backup directory.

4.2.5 Using a Mounted Fileset from a Controller-Based Snapshot as a Backup Source

Some storage controllers, such as the Compaq HSG80 and HSV110, allow you to create a snapshot that copies the critical RAID metadata for a data set to other media and then employs copy-on-write technology to preserve a point-in-time copy of the original data set. Conceptually this is similar to a fileset clone.

4.2.5.1 Advantages and Disadvantages

Table 4–5 lists the advantages and disadvantages of using a controller-based snapshot as a backup source.

Table 4–5: Using a Mounted Fileset from a Controller-Based Snapshot as a Backup Source

Advantages	Disadvantages
Can serve as single backup source for all filesets in domain.	Requires physical storage to hold two or more copies of data.
Off-line time for the original fileset limited to the time it takes to perform the controller-based snapshot, typically a few seconds.	Complex because both controller and AdvFS commands needed.

4.2.5.2 Ensuring Data and Metadata Integrity

To ensure that the data in the fileset is consistent, follow any procedure detailed in Section 4.1.

If the storage making up the domain consists of only one hardware LUN, no special steps are needed to ensure the integrity of the AdvFS metadata. If the storage spans multiple hardware LUNs, see Section 4.1.2.

4.2.5.3 Preparing the Backup Source

The freeze and thaw steps are only necessary if your domain spans more than one LUN. If you are not sure, perform the commands anyway. They

do not harm the domain. Freezing and thawing a single fileset freezes and thaws the entire domain.

The following example assumes that the data is in a domain that has a fileset named `data` mounted on `/data`. The snapshots created are `/dev/disk/dsk25c` and `/dev/disk/dsk26c`.

1. Freeze the domain. The default freeze time is 60 seconds.

```
# /usr/sbin/freezefs /data
```

2. Execute the commands necessary to create the controller-based snapshot. For examples of these commands using the Compaq HSG80 controller, see the Best Practice *Using StorageWorks HSG80 Controller-Based Cloning and Snapshotting*.

3. Thaw the metadata for the domain.

```
# /usr/sbin/thawfs /data
```

Now your applications can continue to run on the original domain.

4. Create block device special files for the newly cloned LUNs. This should be done on the host that will mount the new backup source. The `/sbin/hwmgrr` command is asynchronous, so it could complete before the block device special files are actually created.

On a stand-alone system, enter:

```
# /sbin/hwmgrr -scan scsi
```

On a cluster, enter:

```
# /sbin/hwmgrr scan component -category scsi_bus -cluster
```

Every time a new LUN is created, a new set of device special files is created.

5. Identify the block device special files.

```
# /sbin/hwmgrr -view devices
```

6. Create the directory and symbolic links necessary to access the new domain on `/dev/disk/dsk25c` and `/dev/disk/dsk26c`.

```
# mkdir /etc/fdmns/snap_domain
# ln -fs /dev/disk/dsk25c /etc/fdmns/snap_domain
# ln -fs /dev/disk/dsk26c /etc/fdmns/snap_domain
```

Do not use the `mkfdmn` command to create the new domain. Doing so causes the snapped data to become inaccessible.

7. Create a directory and mount the data fileset in `snap_domain`.

```
# mkdir /backup
```

If the snapshotted domain and fileset are not mounted on the original host:

```
# mount snap_domain#data /backup
```

If the clone domain and fileset are mounted on the original host:

```
# mount -o dual snap_domain#data /backup
```

The backup can now be performed from the /backup directory.

4.3 Backup Methods and Tools

Once you have created a data-consistent and metadata-consistent backup source, you can back up your data. There are a number of ways to do this:

- Mount the backup source on the same host as the original data set to run the backup. This is required if your backup source is the original fileset or a fileset clone.
- Mount the backup source on a host different from the original data set to run the backup. This relieves the original host from the overhead of the backup processing.
- Back up the contents of your raw LUNs directly from disk to tape or other backing store.

There are many backup and restore products available. AdvFS provides the `vdump` and `vrestore` commands and the remote `rvdump` and `rvrestore` commands. The remainder of this section describes how to use these commands.

4.3.1 Introduction to the `vdump` and `vrestore` Commands

Once you have prepared your backup source, you can use the `vdump` command for back up. 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.

The `dump` and `restore` commands function differently from the `vdump` and `vrestore` commands. They work at the inode level so they can handle only UFS files.

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 user and group 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.

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

4.3.2 Unique Features of the `vdump` Command

The `vdump` command has a number of functions that the UFS `dump` command does not have. With the `vdump` command you can do the following:

- 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. 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 recovers data even if a read error occurs when you restore.
- Handle AdvFS and UFS sparse files without zero fills.

4.3.3 Specifying Backup Level Using the `vdump` Command

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.

User and group quota files and fileset quotas are saved only when you are root user and you specify a 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. Therefore, if you back up your files then move or rename them, change the modification date using the `touch` command:

```
touch filename
```

4.3.4 Dumping to Tape Using the `vdump` Command

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. Executing another `vdump` command causes the next saveset to be stored starting at the current tape position.

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

To restore data from a tape containing more than one saveset, see Section 4.3.16.

To locate a tape device on your system, enter the `/sbin/hwmggr` command with the `-view devices` option.

For more information see `hwmggr(8)` and the *Hardware Management* manual.

4.3.5 Dumping to Standard Output Using the vdump Command

When the backup saveset device is the dash (-) character, the `vdump` command writes to standard output. Therefore, 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.

4.3.6 Dumping Subdirectories Using the vdump Command

You can selectively back up individual subdirectories of a fileset. Specify the subdirectory by using the `vdump` command with the `-D` option. Without the `-D` option, if you specify a subdirectory instead of a fileset from 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.

4.3.7 Dumping to Files or Disk Partitions

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, the day's files are backed to a fileset called `projects.Monday`. The fileset is mounted on `/projects`.

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

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

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

You can also store your saveset on an empty partition. However, the saveset is not 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
```

To examine the devices on your system, enter the `/sbin/hwmggr` command with the `-view devices` option.

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 so you lose part of your data. 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 do 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` command with the `-z` option. 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 the *System Administration* manual for more information.

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

4.3.8 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` command with the `-C` option 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.

4.3.9 Dumping with Error Protection

You can use the `vdump` command with the `-x` option 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 is a trade-off because extra storage is needed to provide the added security.

- 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.

4.3.10 Listing Saved Files in a vdump Saveset

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

4.3.11 Backing Up a System Disk Using the clonefsset and vdump Commands— an Example

This example illustrates a level 0 backup of a disk where the `clonefsset` command is used to create the source of the backup information. It assumes that `dsk6`, the backup disk, is the same size disk as `dsk0`, the disk being backed up. In this example the root domain, named `root_domain`, is on `/dev/disk/dsk0a`. The `usr` domain, named `usr_domain`, is on `/dev/disk/dsk0g` and the swap partition is on `/dev/disk/dsk0b`. The `/var` fileset is in `usr_domain`.

It is a good idea to shut down and reboot the system in single-user mode for the duration of fileset clone creation. This ensures that there is no activity on `/`, `/usr`, and `/var`. The backup procedure can be done in multiuser mode, but it is highly recommended that it be done when the system activity is low.

1. Copy the disk label directly from the current system disk and restore it to the new system disk.

```
# disklabel -r dsk0 > /tmp/backupdisklabel
# disklabel -R -r -t advfs dsk6 /tmp/backupdisklabel
# rm /tmp/backupdisklabel
```

2. Create the backup domains and filesets and mount them.

```
# mkfdmn -F -r /dev/disk/dsk6a root_domain_backup
# mkfset root_domain_backup root
```

```
# mkfdmn -F /dev/disk/dsk6g usr_domain_backup
# mkfset usr_domain_backup usr
# mkfset usr_domain_backup var
# mkdir /root_backup /usr_backup /var_backup
# mount root_domain_backup#root /root_backup
# mount usr_domain_backup#usr /usr_backup
# mount usr_domain_backup#var /var_backup
```

3. Create the fileset clones for the filesets and mount them.

```
# clonefset root_domain root root_clone
# clonefset usr_domain usr usr_clone
# clonefset usr_domain var var_clone
# mkdir /clones /clones/root /clones/usr /clones/var
# mount root_domain#root_clone /clones/root
# mount usr_domain#usr_clone /clones/usr
# mount usr_domain#var_clone /clones/var
```

4. Use the `vdump` command to dump the clones and the `vrestore` command to restore the filesets.

```
# vdump -0f - /clones/root | vrestore -xf - -D /root_backup
# vdump -0f - /clones/usr | vrestore -xf - -D /usr_backup
# vdump -0f - /clones/var | vrestore -xf - -D /var_backup
```

5. Change the backup `/etc/fdmns` directory to point to the correct disk.

```
# rm -rf /root_backup/etc/fdmns/root_domain
# mv /root_backup/etc/fdmns/root_domain_backup\
  /root_backup/etc/fdmns/root_domain
# rm -rf /root_backup/etc/fdmns/usr_domain
# mv /root_backup/etc/fdmns/usr_domain_backup\
  /root_backup/etc/fdmns/usr_domain
```

6. Edit the `/etc/sysconfigtab` file to point to the new swap file:

Search for the entry that sets the swap device:

- a. Search for the entry that sets the swap device.

```
swapdevice=/dev/disk/dsk0b
```

- b. Replace it with the new swap device name.

```
swapdevice=/dev/disk/dsk6b
```

7. Shut down the system and reboot from `dsk6` to verify the newly created backup.

If you are creating `dsk6` for backup and plan to return to the original root domain, shut down and reboot from `dsk0`.

4.3.12 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 -of rachem:/dev/tape/tape0 /sar
```

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

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

You cannot back up user and group quota files and fileset quotas with remote backup commands.

The `rvdump` and `rvrestore` commands do not recognize the dash (`-`) character to specify standard output. You must specify the output device.

4.3.13 Unique Features of the `vrestore` Command

The `vrestore` command restores files by processing the blocks from a saveset created by 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.

The `vrestore` command performs a number of activities that the UFS `restore` command does not. The command allows you to do the following:

- Display the current `vrestore` version number.
- Display the source directory path.
- List the saveset structure, that is, the files that were saved.
- Display error messages only. Information messages are not 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.

4.3.14 Restoring Files Using the `vrestore` Command

You can use the `vrestore` command 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.

Run the `vrestore` command with the `-t` option to display the names and sizes of all files in the saveset. The restore operation is not performed.

You can also display the files and directories in the saveset by running the `vrestore` command with the `-i` (interactive) option. You can then 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.

4.3.15 Restoring Quotas Using the `vrestore` Command

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 no UFS analog exists for AdvFS fileset quotas. You must be root user to restore quotas.

4.3.16 Restoring from a Tape with Many Savesets Using the `vrestore` Command

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` command with the `-i` option. 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` command with the `-x` option 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: Fri Apr 26 15:27:36 2002
```

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

5

Optimizing and Tuning the AdvFS File System

You can configure and tune your AdvFS file system in many ways. Some of the tuning functions are available through a graphical user interface. (See Appendix E.) The *System Configuration and Tuning* manual provides detailed information on tunable parameters for AdvFS.

This chapter covers the following:

- Ways to monitor system performance (Section 5.1)
- Improving performance by disabling frag files (Section 5.2)
- Ways to improve transaction log file performance (Section 5.3)
- Data cache tuning (Section 5.4)
- Methods to improve data consistency in the event of a system failure (Section 5.5)
- Direct I/O as a method of improving data transfer (Section 5.6)
- System attributes that might be changed to improve performance (Section 5.7)
- The `vfast` utility, which improves system throughput (Section 5.8)
- Defragmenting a domain (Section 5.9)
- Defragmenting a single file (Section 5.10)
- Balancing the distribution of data (Section 5.11)
- Migrating a file to another volume to improve performance (Section 5.12)
- AdvFS file striping (Section 5.13)
- Moving filesets to reduce the strain on system resource (Section 5.14)
- Controlling the level of a domain panic (Section 5.15)

See the *System Configuration and Tuning* manual and Chapter 1 for more detailed information about allocating domains and filesets effectively. See also Section 2.3.3.1 for an explanation of Version 5 directories, which have indexed directories to improve file access.

5.1 Monitoring Performance

The first step in optimizing a system is to understand how it is performing. You can gather performance information in a number of ways:

- 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 the `iostat` output 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 Menu utility called Monitoring and Tuning — View Input/Output (I/O) Statistics (see Appendix A) 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 the buffer cache, volume reads/writes, the bitfile metadata table (BMT) record, and other parameters for unusual activity. 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, visit the following web site:

<http://www.tru64unix.compaq.com/collect/>

5.2 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. Files or ends of files are stored in the frag file for the fileset, as explained in Section 1.3.3. Fragging, that is, generating frag files, minimizes wasted space in the fileset. If fragging is turned off, I/O is more efficient, but storage requirements increase. Whether or not you choose to have frag files has no effect on the defragment operation (Section 5.9).

Persistent atomic-write data logging requires that a file not have a frag. (See Section 5.5.2 for more information.)

Filesets store fragments of files in the frag file by default. You can disable them by doing the following:

- At fileset creation use the following command format:

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

- For an existing fileset use the following command format:

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

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

```
# showfsets domain_1 fileset_3
  Id           : 3a3a47cb.000b52a5.2.8006
  Files        :      15,  SLim=    0,  HLim=    0
  Blocks (512) :   13730,  SLim=    0,  HLim=    0
  Quota Status : user=off group=off
  Object Safety: off
  Fraggging    : on
  DMAPAPI      : off
```

Disabling or enabling frags in a fileset does not affect existing files. Frags that already exist continue to exist. If you want to eliminate the frag at the end of an existing file, do the following:

1. Change the frag status for the fileset to the one you want for your file by using the `chfsets` command.
2. Copy the fraggged file to a new file.
3. Delete the original file and rename the new file to the original file name.
4. Optionally, change the frag status for the fileset back to frag with the `chfsets` command. If you do this and append to files in the fileset, you can again create frags.

For example, to remove the frag for the file `taxes` in the `arizona` fileset in the `states` domain:

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

To cause files in a fileset to again use the frag file, run the `chfsets` command with the `-o frag` option. The frag file will return for a file once it has been accessed.

For more information see the Best Practice *Improving AdvFS Performance by Disabling the Frag File*.

5.3 Improving Transaction Log File Performance

Each domain has a transaction log file (Section 1.3.2) that keeps track of fileset activity for all filesets in the domain. This creates a high volume of read/write activity. If the log file resides on a congested disk or bus, or if the domain contains many filesets, system performance can degrade. You can

move the log file so that its activity does not use up the bandwidth of the device on which you have stored your files.

Monitor performance of the volumes in the domain with the SysMan Menu utility called Monitoring and Tuning — View Input/Output (I/O) Statistics (see Appendix A) or with the `iostat` utility. If you have AdvFS Utilities, do one of the following if the volume containing the transaction log file appears to be overloaded:

- Divide the domain into several smaller domains. Because each domain has its own transaction log file, 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.

5.3.1 Moving the Transaction Log File

Moving the transaction log to a faster or less congested volume can speed throughput. Moving the log file can also be useful if you are using Logical Storage Manager (LSM) storage. You can increase reliability by placing your transaction log file on a mirrored volume. Then if the disk containing the transaction log file crashes, the mirrored log can be accessed.

To move the transaction log file to another volume, do the following:

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 `/sbin/advfs/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  6 16:24 2002  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%
```

```

# /sbin/advfs/switchlog region1 2
# showfdmn region1
      Id                Date Created      LogPgs Version Domain Name
31bf51ba.0001be10 Wed Feb  6 16:24 K bytes  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%

```

5.3.2 Isolating the Transaction Log File

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, do the following:

1. Make sure the domain is inactive. If not, any writes to a file can cause storage to be allocated on the volume you are trying to isolate.
2. Add a small partition (volume) to the domain for which you are going to isolate the log file. This is where the log file will be placed.

Remember that the I/O load of other partitions on this 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.

3. Use the `switchlog` command to move the log file to the new volume.
4. Use the `showfdmn` command to determine the number of free blocks on the volume with the log file.
5. With the information from the `showfdmn` command, use the `dd` command to build a dummy file of the right size.
6. Migrate the dummy file to the volume containing the log file. This fills the volume completely leaving 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
# /sbin/advfs/switchlog sales 2

```

```
# showfdmn sales
      Id                Date Created      LogPgs Version Domain Name
312387a9.000b049f Thu Mar 14 14:24 2002  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
```

5.4 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 some of the memory used by the file system and reissue the needed memory to the resource requiring it.

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

- 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 the *System Configuration and Tuning* manual 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 5.6.)

5.5 Improving Data Consistency

The method you choose to write data to a file can affect what is saved if a machine fails. You can make two independent choices:

- Whether or not I/O is synchronous
- Whether or not, in addition to the metadata that is written, file data is also written to the transaction log (atomic-write data logging)

5.5.1 Controlling I/O

Write requests, by default, are cached; that is, data is written to the **buffer cache** and not immediately to disk. You can choose how to synchronize I/O writes to a file to balance performance against improved data consistency in the event of a crash.

5.5.1.1 Asynchronous I/O

Asynchronous I/O, the default, generally gives the highest throughput. It combines multiple writes to the same page 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 write can be combined with other contiguous pages in the buffer cache. This enables 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 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 a file. File sizes and locations on disk are consistent, but if the crash occurred before data was written to disk, the user data from recent writes might be out of date. The risk of old data is a trade-off for the increased throughput gained by using asynchronous I/O.

5.5.1.2 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. 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. Also, because the application, not the file system, determines when the data is flushed to disk, the likelihood of consolidating I/Os might be reduced if synchronous write requests are small.

5.5.1.3 Turning Synchronous I/O On and Off

Asynchronous I/O is the default caching method. To turn synchronous I/O on and off for a file, use the `chfile` command with the `-l` option or the `O_SYNC` or `O_DSYNC` flag to the `open()` system call. (See the *Programmer's Guide* and `open(2)`.) Because you cannot use the `-l` and `-L` options of the `chfile` command together, if you plan to use the `chfile` command to control atomic-write data logging, use the system call to activate synchronous I/O. See Section 5.5.2.5 for information on turning atomic-write data logging on and off.

To force all files within a fileset to employ synchronous I/O, use the `mount` command with the `-o sync` option:

```
mount -o sync filename
```

To force all applications accessing a file to employ synchronous I/O, use the `chfile` command with the `-l on` option:

```
chfile -l on filename
```

To turn off synchronous processing and return the file to asynchronous I/O, execute the `chfile` command with the `-l off` option:

```
chfile -l off filename
```

5.5.2 Enabling Atomic-Write Data Logging I/O

Atomic-write data logging writes user data (in addition to the normally logged metadata) to the log file so data is consistent in the event of system failure. Either metadata and file data are written disk or they are not. The I/O method that atomic-write data logging uses depends on whether your I/O has been set to asynchronous (Section 5.5.1.1) or synchronous (Section 5.5.1.2).

Two types of atomic-write data logging are available: persistent (Section 5.5.2.3) and temporary (Section 5.5.2.4). Persistent data logging remains in effect across mounts and unmounts. Temporary data logging is activated for the duration of the mount. You can check the logging status of a file by using the `chfile` command with no options.

Data logging incurs a performance cost because data as well as metadata is written to the transaction log file. This increases the amount of traffic to the log and doubles the I/O for each user write. Grouping files or filesets for which you plan atomic-write data logging into a single or a few domains reduces the burden on other more performance-sensitive domains.

5.5.2.1 Asynchronous Atomic-Write Data Logging I/O

Asynchronous atomic-write data logging I/O is similar to asynchronous I/O except that the user data written to the buffer cache is also written to the log file for each write request. This is done in 8K byte increments. The extra write of the data to the log file ensures data consistency in the event of a crash but can degrade throughput compared with only using asynchronous I/O. If you are using asynchronous I/O and set atomic-write data logging, your I/O is asynchronous atomic-write data logging.

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 8K byte increment of a write is atomic. It 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. Out-of-order disk writes, which might cause inconsistencies in the event of a crash, can never occur.

5.5.2.2 Synchronous Atomic-Write Data Logging I/O

Synchronous atomic-write data logging I/O is similar to asynchronous atomic-write data logging I/O except that the logged data is flushed from the buffer cache to disk before the write request returns to the calling application. Throughput might be degraded compared with using asynchronous atomic-write data logging I/O, because the write does not return until after the log-flushing I/O is complete. If you are using synchronous I/O and set atomic-write data logging, your I/O is synchronous atomic-write data logging.

If a crash occurs while a write is in progress, the data is recovered from the log file when the fileset is remounted. Like asynchronous atomic-write data logging I/O, 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 8K byte increment of a write is either completely written to disk or is not written to disk.

The benefit of synchronous atomic-write data logging is the guarantee of data consistency when a crash occurs after a write call returns to the application. On reboot, the log file is replayed and the user's entire write request is written to the appropriate user data file. In contrast, asynchronous atomic-write data logging guarantees the consistency of only 8K byte increments of data after the write call returns.

5.5.2.3 Persistent Atomic-Write Data Logging

Persistent atomic-write data logging sets an on-disk flag so that the logging persists for the file across mounts and unmounts of the fileset. The choice of whether logging is asynchronous (Section 5.5.1.1) or synchronous (Section 5.5.1.2) depends on how you have set your I/O.

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

```
chfile -L on filename
```

```
chfile -L off filename
```

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. It operates on files with frags. See Section 5.5.2.4.
- 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` command with the `-o nofrag` option, files with existing frags still contain frags. To remove a frag from a file, see Section 5.2.

You cannot use the `-l` and `-L` options of the `chfile` command together to set synchronous I/O and atomic-write data logging. However, if you activate persistent atomic-write data logging on a file by using the `chfile` command with the `-L on` option, you can then 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*.)

Files that use persistent atomic-write data logging cannot be memory mapped through the `mmap` system call. See Section 6.4.10 for information on conflicting file usage.

5.5.2.4 Temporary Atomic-Write Data Logging

Temporary atomic-write data logging sets an in-memory flag for a fileset so that logging persists for the duration of the mount. The choice of whether the data logging is asynchronous or synchronous depends on how you have set your I/O. See Section 5.5.1 for instructions on how to control I/O mode.

Use the `mount` command with the `-o adl, sync` option to set an in-memory flag that activates temporary atomic-write data logging in a fileset for the duration of the mount. Files that have frags can use temporary atomic-write data logging. 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` command with the `-L off` option to turn off persistent atomic-write data logging. All applications that have the file open are affected.

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.

5.5.2.5 Turning Atomic-Write Data Logging On and Off

Whether the data logging is asynchronous (Section 5.5.1.1) or synchronous (Section 5.5.1.2) depends on your I/O setting.

Table 5–1 summarizes the options for turning atomic-write data logging on.

Table 5–1: Turning Atomic-Write Data Logging On

Type of Data Logging	Command	Notes
Persistent (file)	<code>chfile</code> command with the <code>-L on</code> option or <code>fcntl()</code> function	Cannot be activated on files with frags. Cannot use <code>chfile</code> command if <code>chfile</code> command used to control synchronous I/O. See Section 5.5.2.3.
Temporary (fileset)	<code>mount</code> command with the <code>-o adl</code> option for asynchronous or the <code>-o adl, sync</code> option for synchronous	Can activate on files with frags. Persistent atomic-write data logging takes precedence. See Section 5.5.2.4.

Table 5–2 summarizes the options for turning atomic-write data logging off.

Table 5–2: Turning Atomic- Write Data Logging Off

Type of Data Logging	Command	Notes
Persistent (file)	<code>chfile</code> command with the <code>-L off</code> option or <code>fcntl()</code> function	Cannot use <code>chfile</code> command if <code>chfile</code> command used to control synchronous I/O. See Section 5.5.2.3.
Temporary (fileset)	Automatically turns off on fileset unmount or <code>fcntl()</code> function	Any open file can call <code>fcntl()</code> function. See Section 5.5.2.4.

5.6 Improving Data Transfer Rate with Direct I/O

Direct I/O mode bypasses caching; it synchronously reads and writes data from a file without copying the data into a buffer 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, but it offers no guarantee of data consistency in the event of a system crash.

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.

You can only open a file for direct I/O if it is not opened for atomic-write data logging (Section 5.5.2) or it is not memory mapped (Section 6.4.10). 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 its previous mode, once you initiate direct I/O for a file, its mode is direct I/O and remains so until the last close of the file.

You can use the `fcntl()` function to determine if a 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.

5.7 Changing Attributes to Improve System Performance

You can change a number of attributes to improve system performance. The *System Configuration and Tuning* manual details the significance of each attribute and the trade-offs engendered when they are changed. See `sysconfig(8)` for more information. To improve AdvFS performance, the following actions might be useful:

- Increase the dirty-data caching threshold.

Dirty or modified data is data that was written by an application and cached but has 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` command with the `-t` option 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.

To form an I/O of the preferred transfer size of the device, AdvFS coalesces cache pages in memory that are physically contiguous on the disk. The preferred size is determined by the device driver and depends on the underlying storage configuration but is typically 128 or 256 blocks. If you are using LSM, you can change the stripe width, which can result in a larger transfer size. You can adjust the preferred 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 the value of `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.

5.8 Improving Operating System Throughput with the `vfast` Utility

The `vfast` utility is a background process that operates on the files in a domain that are actively opening and closing. The utility performs a number of optimizing functions, such as:

- Reducing file fragmentation
- Balancing volume free space
- Equalizing the I/O load

You must have root user privilege to run the `vfast` utility. You can turn `vfast` processing on or off with the `activate` and `deactivate` options. With the `suspend` option you can turn off `vfast` processing but continue to gather internal statistics. Using the `status` option, you can display the current `vfast` configuration, operational statistics, and processing options for the domain.

Most `vfast` processing occurs when devices have no other system I/O, so running `vfast` does not generally degrade performance. You can limit the share of system I/O that the utility uses with the `-o percent_ios_when_busy=` option. The default is 1% of the known storage device I/O bandwidth.

Some utilities, such as `umount`, `rmvol`, and `rmfset`, suspend `vfast` operations temporarily while they run. When the utilities finish, `vfast` is returned to its prior state.

Because `vfast` defragments and balances domains, the traditional AdvFS defragment and balance utilities cannot be used if `vfast` is activated with the `-o defragment=`, `-o balance=`, or `-o topIObalance=` operations enabled.

The `vfast` command with the `-o defragment=enable` option dynamically consolidates free space, reduces file fragmentation, and makes files more contiguous. Although the AdvFS file system attempts to store file data in contiguous blocks on disk, if contiguous blocks are not available to accommodate the new data, the system spreads data over non-contiguous blocks. This fragmentation degrades the read/write performance because many disk addresses must be examined to access a file. The `vfast` utility operates to minimize this degradation by moving fragmented files to contiguous disk blocks. Files might be relocated during consolidation.

Only files that are opening and closing are added to the `vfast` cache of fragmented files. To defragment files with `vfast`, at least one fileset in the domain must be mounted read/write. If other filesets in the domain are not mounted for writing, then the fragmented files in these filesets are not defragmented unless the filesets were previously mounted as writable and there are still files from these filesets in the `vfast` cache waiting to be processed. Run the `vfast` command with the `-l extents` option to display files queued for defragmentation. If there are no files in the `vfast` cache, no defragmentation or free-space balancing takes place.

To defragment a complete domain, you must open and close all the files in the domain. For each fileset in the domain, do the following:

1. Mount the fileset and change to its directory.
2. Execute the following command:

```
# find ./ -name \* >/dev/null
```
3. Because the `vfast` cache can fill up, all the files may not be defragmented in a single pass. To completely defragment a domain, you might need to open and close the files more than once. To check progress of the defragmentation, enter the following:

```
# vfast -L extents domain_name
```

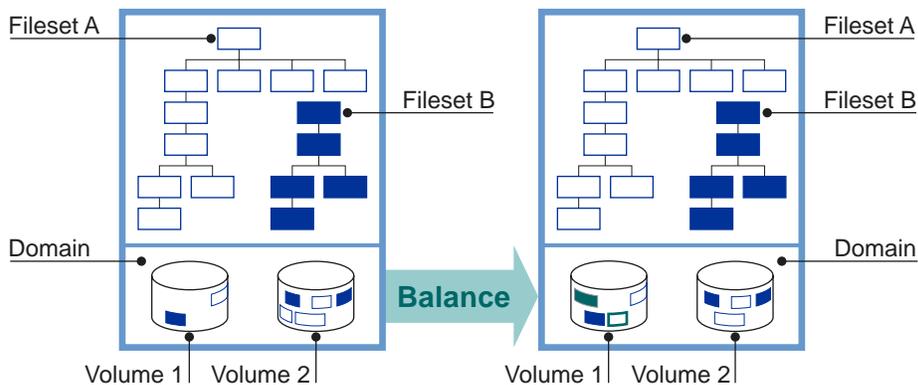
You cannot control the placement of files when `vfast` defragments a multivolume domain. To identify where a file is stored, execute the `showfile` command. If you want to move a particular file to a different volume, use the `migrate` command. (See Section 5.12.)

You can stop defragmenting at any time by setting `-o defragment=disable`. Discontinuing the process does not damage the file system. Files that have been defragmented remain in their new locations.

If files are enabled for direct I/O (Section 5.6), the `vfast` utility defragments these files unless the `-o direct_io=` option is set to `disable`.

If you have enabled the `-o defragment=` option, activating `-o balance=` as well causes the `vfast` utility to distribute files between volumes of a multivolume domain to equalize the I/O load. The utility moves files from one volume to another, as illustrated in Figure 5-1, until the percentage of used space on each volume in the domain is as equal as possible. Only files queued for defragmentation are used to balance domains.

Figure 5–1: Balancing a Domain



Only files in need of defragmentation are balanced to equalize the free space across volumes. If there are no fragmented files in the domain, then `vfast` does not perform any balancing. Furthermore, only files actively being closed are checked for fragmentation, so files are ignored if they have not been accessed since the last mount of the fileset.

Enabling the `-o topIObalance=` option causes `vfast` to first monitor I/O (for an interval set by the `-o ss_steady_state=` option) to determine which volumes are experiencing the heaviest I/O load. The utility then continuously distributes files with high I/O among the volumes of the domain to balance the load. However, if files are enabled for direct I/O, `topIObalance` ignores these files because I/O to these files bypasses `vfast` statistics collection.

If your domain is striped in any way (hardware RAID, LSM striping, AdvFS file striping), do not use the `-o topIObalance=` option. The utility cannot effectively distribute the I/O load in this configuration.

If a system node is a cluster member, some files may not be processed initially because the cluster file system (CFS) caches files and processing cannot occur until the cache is flushed.

5.8.1 Choosing to Run the `vfast` Utility

To start `vfast` processing for a domain, execute the `vfast` command with the `activate` option. Once you have initiated processing, you can choose the types of activities the utility performs by executing additional `vfast` commands.

To identify the amount of fragmentation in your domain without enabling `vfast`, use the `-L extents` option. If the average number of extents or the number of extents per file that have extents is high, running the utility

might be helpful. If you think the domain has excessive fragmentation, run `vfast` with the `-o defragment=` option.

To determine the layout of your domain, execute the `showfdmn` command. Look at the `% Used` field to determine if the files are evenly distributed among volumes. If they are not, and you have enabled defragmenting, run the `-o balance=` option on the domain.

Use the `vfast` command with the `-l hotfiles` option to identify the most actively paging files by volume and domain. The `-L hotfiles` option displays a volume distribution summary by domain.

5.8.2 Examples of the `vfast` Utility

The examples in this section use the `vfast` options to control the functionality of the utility.

The following example initiates selected `vfast` functionality:

```
# vfast activate user_dmn
# vfast -o defragment=enable user_dmn
# vfast -o balance=enable user_dmn

# vfast -o topIObalance=enable user_dmn
# vfast -o percent_ios_when_busy=20 user_dmn
```

The following example uses the `status` option to determine the amount of defragmentation done to the domain `user_dmn`.

```
# vfast status user_dmn
vfast is currently running
vfast is activated on user_dmn
vfast defragment:      enabled
vfast balance:        enabled
vfast top IO balance:  enabled
Options:
  Direct IO File Processing: enabled
  Percent IOs Allocated to vfast When System Busy: 20%
  Default Hours Until Steady State: 24; Hours remaining: 0
  Total Files Defragmented: 3331
  Total Pages Moved for Defragment: 278440
  Total Extents Combined for Defragment: 21
  Total Pages Moved for Balance: 0
  Total Files Moved for Volume IO Balance: 0
  Total Pages Moved for Volume Free Space Consolidation: 50607
```

The following example uses the `-l extents` option to display the volumes queued for defragmentation and the volume free space balancing in the domain `user_dmn`:

```
# vfast -l extents user_dmn
user_dmn: Volume 1
```

```

extent
count   fileset/file
  2 user: /u1/obj/BINARY/lp.o
  2 user: /u1/w17/obj/kernel/test_21.o
  3 user: /u1/obj/bs_bitfile_sets.o
  3 user: /u1/w4/itpsa.o
  4 user: /u1/w4/cms_utils.o

```

The following example uses the `-L extents` option to display fragmentation summaries by volume for the domain `user_dmn`.

```

# vfast -L extents user_dmn
user_dmn
  Extents:                46003
  Files w/extents:        45694
  Avg exts per file w/exts:  1.01
  Free space fragments:    18858
                        <100K  <1M   <10M  >10M
  Free space:             21%    41%    31%    7%
  Fragments:             16119  2523  213    3

```

The following example uses the `-l hotfiles` option to look at the most actively paging files and the volumes on which they reside in the domain `user_dmn`.

```

# vfast -l hotfiles user_dmn |more
Past Week

IO Count Volume File
5487993   1   *** a reserved file, tag = -2,-10, BMT
197088    1   *** a reserved file, tag = -2, -7, SBM
147757    1   *** a reserved file, tag = -2, -9, LOG
  2814    1   user: /user1/crl/BINARY/makedep
  1206    1   user: /user1/crl/applications/sequoia.jar
  1005    1   user: /user1/sandboxes/advfs.mod
   402    1   user: /user1/alpha/arch.mod

```

5.9 Defragmenting a Domain

If you are not running the `vfast` utility (Section 5.8), you can run the `defragment` utility to reduce the amount of file fragmentation (Section 1.3.3) in your domain. This utility attempts to make the files more contiguous so that the number of file extents is reduced. Because many disk addresses must be examined to access a fragmented file, defragmenting a domain improves the read/write performance. 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. You can defragment a domain if you have frag files turned on or off. The activities are not related.

The `vfast` utility is preferable for defragmenting a domain because it is optimized for the operating system and runs in the background. You can improve the efficiency of running the `defragment` utility by deleting unneeded files in the domain before running it. Run the `defragment` utility on your domain when you experience performance degradation and then only when file system activity is low.

You can stop defragmenting at any time. Aborting the process does not damage the file system. Files that have been defragmented remain in their new locations.

You cannot control the placement of files during defragmentation of a multivolume domain. Use the `showfile` command to identify where a file is stored. If you want to move a file, execute the `migrate` command. See Section 5.12 for more information about migrating files.

To defragment a domain, all filesets in the domain must be mounted. 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. Use the SysMan Menu utility called Manage an AdvFS Domain (see Appendix A), a graphical user interface (see Appendix E), or enter the `defragment` command from the command line:

```
defragment domain_name
```

You must have root user privileges to defragment a domain. The `defragment` utility cannot be run while the `vfast`, `addvol`, `rmvol`, `balance`, or `rmfset` command is running in the same domain.

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

- The size of the volumes
- The amount of free space available
- The activity of the system
- The configuration of your domain

Because the `defragment` utility creates one 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. However, 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), execute the `defragment` command with the `-N 1` option.

To determine the amount of fragmentation in your domain without starting the utility, run the `defragment` command with the `-v -n` options. 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. 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 due to excessive file fragmentation.

If you find that one file shows high fragmentation, you can defragment that file individually. See Section 5.10 for a discussion of how to defragment a file.

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 `/sbin/advfs/verify` command (Section 6.2.4) 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 8K bytes, the AdvFS page size, you need only run the `defragment` command when the `frag` file for the fileset, called `/mount_point/.tags/1`, is highly fragmented. See Section 5.2 for a discussion of disabling the `frag` file.

If you have the hardware resources and AdvFS Utilities, you can add a volume by using the `addvol` command then remove the original 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 displays the fragmentation of the `accounts_domain` domain and then defragments the domain for a maximum of 15 minutes.

```
# defragment -v -n 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

# 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.

See defragment(8) and the Best Practice *Defragmenting an AdvFS Domain* for more information.

5.10 Defragmenting a File

You can defragment a file without defragmenting the entire domain. You can defragment a single file while the `vfast` utility is running.

To determine if a file is a good candidate for defragmentation, that is, the file has a large number of extents, run the `showfile` command with the `-x` option. If the number of extents is large, do one of the following to decrease fragmentation:

- 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.
 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.

5.11 Balancing a Multivolume Domain

If you are not running the `vfast` utility (Section 5.8), you can run the `balance` utility to distribute the percentage of used space evenly between volumes in a multivolume domain. This improves performance and evens the distribution of future file allocations.

The utility moves files from one volume to another until the percentage of used space on each volume in the domain is as equal as possible. See Figure 5-1. This process is the same as that used by the `vfast` utility. 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, all filesets in the domain must be mounted. Use the SysMan Menu utility called Manage an AdvFS Domain (see Appendix A), a graphical user interface (see Appendix E), 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.

You must have root user privileges to balance a domain. The `balance` utility cannot be run while the `vfast`, `addvol`, `rmvol`, `defragment`, or `rmfset` command is running in the same domain.

To determine if your files are evenly distributed, execute the `showfdmn` command to display domain information. Look at the `% Used` field to determine the files distribution.

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

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 Wed Apr 3 10:50:05 2002 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 Wed Apr 3 10:50:05 2002 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%
```

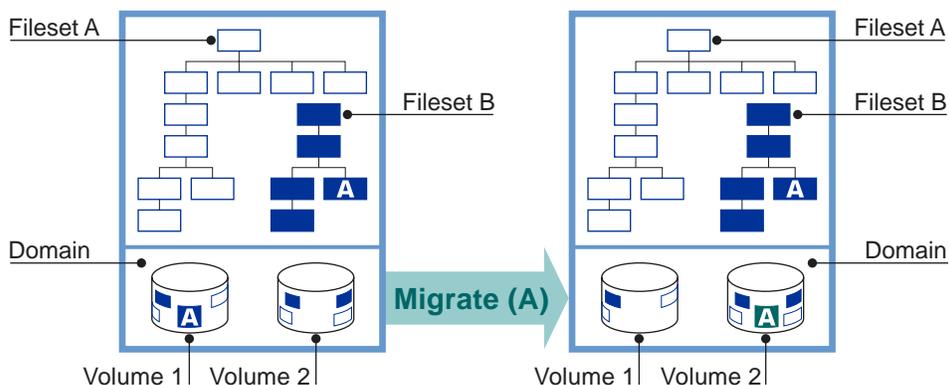
See `balance(8)` for more information.

5.12 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 any volume you choose in the domain. If you have a high performance device, you might want to move an I/O intensive file to it.

The `balance` utility, the `defragment` utility, and the `vfast` utility with the `-o topIObalance=` option migrate files. Only with the `migrate` utility can you choose which files or pages to move and their destinations. You can migrate either the entire file or specific pages. Figure 5–2 illustrates the migration process for file A, which moves from volume 1 to volume 2.

Figure 5–2: Migrating Files



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

```
migrate -d destination_vol_index filename
```

A file that is migrated is defragmented in the process if possible. You can use the `migrate` command to defragment selected files. The `migrate` utility does not evaluate your migration decisions. You can move a striped file segment to a disk where another segment resides, defeating the purpose of striping.

You must have root user privileges to migrate a file. You can perform only one migrate operation at a time on the same file and you can migrate from only one volume at a time.

The following example uses the `showfile` command with the `-x` option to look at the extent map (Section 1.3.3) and the performance of a file called `src`. This file, which belongs to a two-volume domain, is migrated to another volume. It shows a change from 11 file extents to one and a performance efficiency improvement from 18% to 100%. 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.

```
# 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

```

5.13 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, hardware 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 volumes. Combining AdvFS striping with system striping might degrade performance.

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

The form of the AdvFS stripe command is:

```
stripe -n volume_count filename
```

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

You can remove AdvFS striping in your domain by doing one of the following:

- Removing striping from a file
 - Copy the striped file to a file that is not striped. Delete the original.
- Removing the 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 no volume is free of stripes, stripe a new file across the existing volumes and copy the file with the doubled-up segments to the new file.

For more information, see `stripe(8)`.

5.14 Moving a Domain and its Filesets to a New Volume

If you have added a new volume, or if you believe that a fileset or domain is straining system resources, you can move a domain to a different volume. To determine whether to move a domain, look at I/O performance on the device on which it is located. Run the `iostat` utility either from the SysMan Menu utility called Monitoring and Tuning — View Input/Output (I/O) Statistics (see Appendix A), or from the command line (see `iostat(1)`).

If you want to move an entire domain and its fileset to a new volume, do the following:

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. Remove the old domain.
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.

The new domain is created with the new domain version number (DVN) of 4. (See Section 2.3.3.1 for an explanation of domain version numbers.) If you

must retain the DVN of 3 to use on 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 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 -dx - /technical|vrestore -xf - -D /tmp_mnt
# umount /technical
# umount /tmp_mnt
# rmdmn accounts
# rmdir /tmp_mnt
# mv /etc/fdmns/new_accounts/ /etc/fdmns/accounts/
# mount accounts#technical /technical
```

5.15 Controlling Domain Panic Information

Use the `AdvfsDomainPanicLevel` attribute to choose whether to have crash dumps created when a domain panic occurs. A common cause for domain panics is I/O errors from a device. AdvFS must panic a domain if it cannot write metadata. In the current implementation, an I/O error of this type does not cause a crash dump to be created unless the system crashes.

To force a crash dump, set the value of the attribute as follows:

- 0 – Do not create crash dumps.
- 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 6.3.1 for information about recovering from a domain panic.

6

Troubleshooting

This chapter examines recovery procedures for problems that might occur and actions that can be taken to fix or avoid future problems. The chapter covers the following:

- Recovery of a single user file (Section 6.1)
- System recovery (Section 6.2)
- Recovery and restoration procedures for a number of common problems (Section 6.3)
- Preventative maintenance strategies (Section 6.4)

6.1 Recovering a User File

If you do not plan for this in advance, there is no convenient way to recover a file that has been deleted. You can plan for access to files that have been removed in three ways:

- Back up the directory with the `vdump` command and restore the file with the `vrestore` command. You must have write permission for the directory where the backup is saved. See Chapter 4 for more information.
- Set up a fileset clone, preferably each day, to access files that have changed that day. You must have root privileges to create a clone. See Section 2.4.10 for more information.
- Set up a trashcan for the directory. This feature is under user control. Root privileges are not required. See Section 2.6 for more information.

If a file has been removed and there is no backup, trashcan or clone, there is no way to recover the file. If a file is corrupted, however, running the `/sbin/advfs/salvage` command may recover some of its contents. See Section 6.2.6 for directions.

6.2 General Recovery Procedures

The following recovery steps apply to all domain panics and many types of system crashes. If the cause of the crash appears to be in one domain, the following steps may be needed to regain access to the filesets in the affected domain. Follow as many steps as needed. All might not be necessary. If you are recovering the root domain, see Section 6.3.4.

6.2.1 Saving Data for a Problem Report

Run the `sys_check` utility with the `-escalate` option to create escalation files for reporting problems to your technical support representative. The utility collects the output in the file `/TMPDIR/escalate.tar` file.

You can run the `sys_check` command either from the SysMan Menu utility called Support and Services — Create Escalation Report (Escalation) (see Appendix A), or from the command line.

You must be root user or have appropriate privileges to run the `sys_check` utility. See `sys_check(8)` for more information.

6.2.2 Saving Copies of System Metadata

If it appears that a domain is corrupted or it is otherwise causing problems, run the `/sbin/advfs/savemeta` command to save a copy of the domain's metadata for support personnel to examine. Run the `savemeta` command with the `-f` option to save structure information from the frag files for each fileset. The utility saves the log file, bitfile metadata table, each volume's storage bitmap, the domain's root tag file, and the fileset tag files.

For example, to save metadata from `domain_1` on `/tmp/saved_domain_1` enter:

```
# /sbin/advfs/savemeta domain_1 /tmp/saved_domain_1
```

To save metadata from the frag file in the fileset `fsetN` on `/tmp/saved_fsetN` enter:

```
# /sbin/advfs/savemeta domain_1 -f fsetN /tmp/saved_fsetN
```

You must be root user to run this command. See `savemeta(8)` for more information.

6.2.3 Saving Undamaged Filesets

It is a good idea to back up as many filesets as you can in the affected domain. (See Chapter 4 for a description of backup methods.) You can use this backup, other recent backups, and the output of the recovery utilities (`/sbin/advfs/verify`, `/sbin/advfs/fixfdmn`, and `/sbin/advfs/salvage`) to recreate your domain.

6.2.4 Verifying File System Consistency

To ensure that metadata is consistent, run the `/sbin/advfs/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 in the following situations:

- 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 Menu utility called Manage an AdvFS Domain (see Appendix A), or enter the `verify` command from the command line:

```
/sbin/advfs/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` command with the `-F` option. The `-F` option mounts the fileset using the `-d` option of the `mount` command, so 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` command with the `-F` option can cause data corruption.

Under some circumstances the `verify` command does not unmount the filesets. If this occurs, unmount the affected filesets manually and delete the mount points that were created in the `/etc/fdmns` file for the domain.

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. See `swapon(8)` for more information.

If running the `verify` command is successful, it is a good idea to back up your domain in case the event that caused the crash recurs.

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

```
# /sbin/advfs/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 so no further recovery steps are needed. See `verify(8)` for more information.

6.2.5 Fixing On-Disk Metadata Corruptions with the `fixfdmn` Utility

The `/sbin/advfs/fixfdmn` utility is designed primarily to put a domain into a usable (mountable) state by repairing metadata corruptions. Unmount your fileset and run the utility in the following situations:

- If the `/sbin/advfs/verify` utility described in Section 6.2.4 cannot run
- If the `verify` utility detects on-disk (metadata) corruption

- If you have a domain with filesets that cannot be mounted without a domain panic and you do not want to run the `verify` command with the `-F` option

The `fixfdmn` utility scans metadata looking for corruptions and, if enough metadata is intact, it attempts to correct the corrupt metadata. If not enough viable metadata is available, the utility attempts to bypass the corruption by moving or deleting the corrupt metadata and deleting files as necessary.

The utility does not check or repair the contents of files. If recovering data from a file is your priority or if a large portion of the domain is corrupted, see Section 6.2.6 for information on using the `/sbin/advfs/salvage` utility.

Once you have run the `fixfdmn` utility, remount the filesets. If you can remount your filesets, check that no files have been deleted. If they have, restore them from backup. If you are unable to mount the filesets, run `fixfdmn` with the `-u` option to undo the process so that the `salvage` command can access the unchanged data.

If running the `fixfdmn` command is successful, it is a good idea to back up your domain in case the event that caused the crash recurs. Running the `fixfdmn` command successfully means that you do not need additional recovery procedures.

You can run the `fixfdmn` command with the `-n` option to check the domain and not do any repairs. The utility will print system messages, but changes will not be written to disk.

6.2.6 Salvaging File Data from a Corrupted Domain

Run the `/sbin/advfs/salvage` utility to recover the files in the domain. The 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.

If you have run the `/sbin/advfs/fixfdmn` command and it was not successful, before you run the `salvage` command, you must first run the `fixfdmn` command with the `-u` option to undo the changes that the utility has made.

You can access `salvage` functionality through the SysMan Menu utility called Manage an AdvFS Domain (see Appendix A) or you can enter the `salvage` command from the command line:

```
/sbin/advfs/salvage domain_name fileset_name
```

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 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 log every file that is encountered.

The `salvage` command places the recovered files in directories named after the filesets. The utility creates a `lost+found` directory for each fileset and puts files in it 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 in a `tar` format by specifying the `-F tar` option.

6.2.6.1 Recovering File Data

If you have a recent backup source, use the `-d` option of the `/sbin/advfs/salvage` utility to recover only the data that has changed after the date of the backup. This limits the amount of information the `salvage` utility must process. If you do not have a backup source, the `salvage` utility can look at all files in the domain you name.

If your system does not have enough space to hold the recovered files, you can recover data to tape and then write it back to your original disk location. You can also execute the `salvage` command using the `-F` and `-f` options to recover data in a `tar` format.

The process is as follows:

1. Unmount all filesets in the corrupted domain.
2. If you are saving the salvaged data to tape, install a tape on a local drive. See the *Hardware Management* manual for more information.
3. If you are not saving salvaged data in `tar` format, create a temporary domain and fileset to hold the recovered information and mount the fileset.
4. Run the `salvage` command.
5. View the `salvage.log` file to ensure that all necessary files were recovered. If files are missing, you will need to use a backup source for these files.
6. Recreate the domain and mount-point directory. Create filesets and mount them.

7. Restore files from backup.
8. Restore the salvaged files from either the temporary domain or the tar file.

The following example salvages data to disk. In this example the corrupted domain is called PERSONNEL and contains the fileset personnel_a-e mounted at /personnel_a-e. The original domain is on volume /dev/disk/dsk12c and the salvage command places output on /dev/disk/dsk3c. Because a backup tape, /dev/tape/tape0, is available, only files from the PERSONNEL domain that were modified after 1:30 PM on December 5, 2001 (the date of the last backup) are extracted from the damaged domain. The domain is restored back to the original volume.

```
# umount /PERSONNEL
# mkfdmn /dev/disk/dsk3c RECOVER
# mkfset RECOVER recover_fset
# mkdir /recover
# mount RECOVER#recover_fset /recover
# /sbin/advfs/salvage -d 200112051330 -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: Starting search of all filesets: 09-Dec-2001
salvage: Starting search of all volumes: 09-Dec-2001
salvage: Loading file names for all filesets: 09-Dec-2001
salvage: starting recovery of all filesets: 09-Dec-2001
```

Before running the mkfdmn command, check the salvage.log file to make sure that you have the files you want. If you are missing files, it may be helpful to run salvage inserting an earlier date for the -d option to retrieve additional information. Once you have created a new domain, you cannot run salvage again except with the -S option, as described in Section 6.2.6.3.

```
# mkfdmn /dev/disk/dsk12c PERSONNEL
# mkfset PERSONNEL personnel_a-e
# mkdir /personnel_a-e
# mount PERSONNEL#personnel_a-e /personnel_a-e
# vrestore /dev/tape/tape0 /personnel_a-e
# cp -Rp /RECOVER/personnel_a-e/* /personnel_a-e
# umount /recover
# rmfdmn RECOVER
rmfdmn: remove domain RECOVER [y/n] y
rmfdmn: domain RECOVER removed.
```

6.2.6.2 Recovering 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 6.3.4. Depending on the nature and extent of the root domain corruption, successful file recovery may not be possible.

6.2.6.3 Recovering Data Block by Block

If you ran the `salvage` utility and were unable to recover a large number of files, run the `salvage` command with the `-S` option. This process is very slow because the utility reads every disk block at least once.

Note

If you have accidentally used the `mkfdmn` command on a good domain, running the `salvage` command with the `-S` option 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 from a domain called `PERSONNEL` 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: 07-Oct-2001
salvage: Loading file names for all filesets: 07-Oct-2001
salvage: Starting recovery of all filesets: 07-Oct-2001
```

6.3 Fixing Problems

This section addresses a number of ways to restore your system.

6.3.1 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 (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`.

To recover from a domain panic, perform the following steps:

1. Run the `mount` command with the `-t advfs` option and identify all mounted filesets in the affected domain.
2. Unmount all the filesets in the affected domain.
3. Save the system problem report information (Section 6.2.1).
4. Save the domain metadata. See `savemeta(8)` and Section 6.2.2 for more information.
5. If the problem is a hardware problem, fix it before continuing.
6. Run the `/sbin/advfs/verify` utility (Section 6.2.4) on the domain.
 - If there are no errors, mount all the filesets you unmounted and resume normal operations.
 - If the `verify` command runs but shows errors, run the `/sbin/advfs/fixfdmn` command (Section 6.2.5) to fix metadata corruption.
7. If the failure prevents complete recovery, recreate the domain (Section 6.2.6).

The following example illustrates the commands needed after a domain panic on the `staffb_dmn` domain:

```
# 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
# sys_check -escalate
# /sbin/advfs/savemeta staffb_dmn /tmp/saved_dmn
```

```
# /sbin/advfs/savemeta -f staff3_fs /tmp/saved_staff3
# /sbin/advfs/savemeta -f staff4_fs /tmp/saved_staff4
# /sbin/advfs/verify staffb_dmn
```

You do not need to reboot after a domain panic.

Domain panics preceded by messages that report I/O errors are usually caused by faulty hardware that must be fixed. The following is an example of an I/O error message:

```
AdvFS I/O error:
Domain#Fileset: staffb_dmn#staff3_fs
Mounted on: /usr/staff3
Volume: /dev/disk/dsk6c
Tag: 0x00000006.8001
Page: 56
Block: 1200
Block count: 128
Type of operation: Write
Error: 19
EEI: 0x6200 (Advfs cannot retry this)
AdvFS initiated retries: 0
Total AdvFS retries on this volume: 0
I/O error appears to be due to a hardware problem.
Check the binary error log for details.
To obtain the name of the file on which
the error occurred, type the command:
  /sbin/advfs/tag2name /usr/staff3/.tags/6
```

If you have recurring domain panics that do not indicate I/O errors, first try running the `fixfdmn` utility (Section 6.2.5). If domain panics still occur after running `fixfdmn`, try adjusting the `AdvfsDomainPanicLevel` attribute (Section 5.15) to facilitate debugging.

6.3.2 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. Two errors are common:

- A disk is mislabeled on a RAID array.
- A volume upon which an AdvFS domain resides was reduced from its original size.

If you have AdvFS Utilities, and if the domain consists of multiple volumes with enough free space to remove the offending (flagged) volume, you do not need to remove your filesets. However, it is a good idea to 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. If you do not have enough free space on the remaining volumes, use the `addvol` command to add additional space before removing the volume.
2. Correct the disk label of the offending 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_a`) within the `data5` domain is mislabeled, you can migrate your files from that volume (automatic with the `rmvol` command), then move them back after you have corrected and restored the volume. Note that the first `addvol` step is necessary only if you need space in the domain for the information on the volume that you are removing.

```
# addvol /dev/disk/dsk3c data5
# rmvol /dev/disk/dsk2c data5
# disklabel -z dsk2
# disklabel -rw dsk2 disk_a
# addvol /dev/disk/dsk2c data5
# balance data5
```

If you do not have AdvFS Utilities, the process is as follows:

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. Restore the filesets from the backup.

For example, if `/dev/disk/dsk1c` (on a device here called `disk_a`) containing the `data3` domain is mislabeled:

```
# vdump -0f -u /data3
# rmfdmn data3
# disklabel -z dsk1
# disklabel -rw dsk1 disk_a
# mkfdmn data3
# mkfset data3 data3fset
# mount data3#data3fset /data3
# vrestore -xf - /data3
```

6.3.3 Restoring the /etc/fdmns Directory

AdvFS must have a current `/etc/fdmns` directory to mount filesets. (See Section 2.3.1 for more information.) 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 (Section 6.3.3.1) or with the `/sbin/advfs/advscan` command (Section 6.3.3.2). 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.

6.3.3.1 Reconstructing the /etc/fdmns Directory Manually

If you choose to reconstruct the directory manually, you must know the name of each domain and its associated volumes.

Caution

Do not run the `mkfdmn` command to reconstruct an existing domain. Executing the command destroys the data on that volume. If you have destroyed your files, you might be able to recover some files by running the `salvage` utility with the `-S` option.

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). The order of creating the links in these examples does not matter. The domains are `domain1` on `/dev/disk/dsk1c` and `domain2` on `/dev/disk/dsk2c`.

```
# mkdir /etc/fdmns
# mkdir /etc/fdmns/domain1
# cd /etc/fdmns/domain1
# ln -s /dev/disk/dsk1c .
# mkdir /etc/fdmns/domain2
# cd /etc/fdmns/domain2
# ln -s /dev/disk/dsk2c .
```

The following example reconstructs one multivolume domain. The domain1 domain contains the three volumes /dev/disk/dsk1c, /dev/disk/dsk2c, and /dev/disk/dsk3c.

```
# mkdir /etc/fdmns
# mkdir /etc/fdmns/domain1
# cd /etc/fdmns/domain1
# ln -s /dev/disk/dsk1c .
# ln -s /dev/disk/dsk2c .
# ln -s /dev/disk/dsk3c .
```

6.3.3.2 Reconstructing the /etc/fdmns Directory Using the advscan Command

Use the /sbin/advfs/advscan command to determine which partitions on a disk or which Logical Storage Manager (LSM) volumes are part of an AdvFS domain if you have moved disks to a new system, if device numbers have changed, or if you have lost track of a domain location. You can also use the command to rebuild all or part of your /etc/fdmns directory if you accidentally deleted it, deleted a domain it, or deleted links from a domain's subdirectory. See advscan(8) for more information.

For each domain there are three numbers that must match for the AdvFS file system to operate properly:

1. The number of physical partitions found by the advscan command that have the same domain ID
2. The domain volume count (the number stored in the AdvFS metadata that specifies the number of partitions in the domain)
3. The number of /etc/fdmns links to the partitions, because each partition must be represented by a link

The advscan command attempts to repair the domain if these numbers are inconsistent.

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 number of /etc/fdmns links. The following tables list anomalies, possible causes, and corrective actions that the advscan utility can take.

Table 6–1 describes possible causes and corrective actions if the number of links in the `/etc/fdmns/domain_name` directory does not equal the expected value for the number of partitions and for the domain volume count.

Table 6–1: Incorrect Number of Links

Number of Links	Possible Cause	Corrective Action
Too low	The <code>addvol</code> command terminated early or a link in <code>/etc/fdmns/domain_name</code> was manually removed.	If the domain is activated before running the <code>advscan</code> command with the <code>-f</code> option, and the cause of the mismatch is an interrupted <code>addvol</code> command, the situation is corrected automatically. Otherwise, the <code>advscan</code> utility adds the partition to the <code>/etc/fdmns/domain_name</code> directory.
Too high	The <code>rmvol</code> command terminated early or a link in <code>/etc/fdmns/domain_name</code> was manually added.	If the domain is activated and the cause of the mismatch is an interrupted <code>rmvol</code> command, the situation is corrected automatically. If the cause is a manually added link in <code>/etc/fdmns/domain_name</code> , systematically try removing different links in the <code>/etc/fdmns/domain_name</code> directory and activating the domain. The number of links to remove is the number of links in the <code>/etc/fdmns/domain_name</code> directory minus the domain volume count displayed by the <code>advscan</code> command.

Table 6–2 shows possible causes and corrective actions if the domain volume count does not equal the expected value for the number of partitions and for the number of links in the `/etc/fdmns/domain_name` directory.

Table 6–2: Incorrect Domain Volume Count

Domain Volume Count	Possible Cause	Corrective Action
Too low	Cause unknown.	Cannot correct; run the <code>/sbin/advfs/fixfdmn</code> command to put your domain in a usable state.
Too high	The <code>addvol</code> command terminated early and the partition being added is missing or was reused.	Cannot correct; run the <code>salvage</code> utility to recover as much data as possible from the remaining volumes in the domain.

Table 6–3 shows possible causes and corrective actions if the number of partitions does not equal the expected value for the domain volume count and for the number of links in the `/etc/fdmns/domain_name` directory.

Table 6–3: Incorrect Number of Partitions

Number of Partitions	Possible Cause	Corrective Action
Too low	Partition missing.	Cannot correct; run the <code>fixfdmn</code> command to put your domain in a usable state.
Too high	The <code>addvol</code> command terminated early.	Rerun the <code>addvol</code> command.

In the following example no domains are missing. The `advscan` command scans devices `dsk0` and `dsk5` for AdvFS partitions and finds two partitions: `dsk0c` and `dsk5c`. The domain volume count reports two, and two links are entered in the `/etc/fdmns` directory.

```
# /sbin/advfs/advscan dsk0 dsk5
Scanning disks  dsk0 dsk5
Found domains:
usr_domain
      Domain Id      2e09be37.0002geb40
      Created        Thu Feb 28 09:54:15 2002
      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. Thus 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. No domain directory 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_name` directories.

If a directory for a partition does not have an entry in the `/etc/fdmns` directory, the partition name is marked with an asterisk (*) and the utility creates a domain name for it. You must then rename the recovered domain to the name of your original domain. In this example, the original domain names were `revenue` and `expenses`.

```
# /sbin/advfs/advscan -r dsk6
Scanning disks dsk6
Found domains:
*unknown*
          Domain Id      2f2421ba.0008c1c0
          Created        Sun Jan 20 13:38:02 2002

          Domain volumes      1
          /etc/fdmns links    0

          Actual partitions found:
                                dsk6a**
unknown*
          Domain Id      2f535f8c.000b6860
          Created        Mon Feb 25 09:38:20 2002

          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
# mv /etc/fdmns/domain_dsk6a /etc/fdmns/revenue
# mv /etc/fdmns/domain_dsk6b /etc/fdmns/expenses

```

6.3.4 Recovering from Data Corruption of an AdvFS root Domain

Catastrophic corruption of your AdvFS root domain typically requires that you recreate your root file system to have a bootable system. If you are running a cluster configuration, see the *System Administration* and the *Cluster Administration* manuals. If your root volume is an LSM volume, see the *Logical Storage Manager* manual. You must be root user to reconstruct the root domain.

An installation CD-ROM or a Remote Installation Service (RIS) server is required. The root domain is recreated on the boot device. If you do not have adequate backup of the root domain, depending on the nature and extent of the corruption, you might be able to recover root files using the `/sbin/advfs/salvage` utility. (See Section 6.2.6 for more information.) The `salvage` utility might also be used to recover files that were modified or created following the most recent backup.

First identify hardware resources.

1. If you plan to boot your system from the installation CD-ROM, determine the name of your CD-ROM drive by entering the following at the System Reference Manual (SRM) console prompt:

```

>>> show device
...
DKA500          RRD47    1206    dka500.4.0.5.0
...

```

In this example the CD-ROM device name is DKA500.

If you plan to boot from a RIS server, determine the name of your network interface device by entering the following:

```

>>> 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.

2. Identify the console boot device name. If your boot device is the default boot device, identify it by entering the following:

```
>>> show bootdef_dev
bootdef_dev      dkb400.4.0.5.1
```

In this example, `dkb400.4.0.5.1` indicates that `dkb400` is the boot device, `dk` indicates that the device is a SCSI disk, `b` indicates that the device is connected to SCSI bus `b`, and `400` indicates that the device's SCSI target ID is 4 and its Logical Unit (LUN) is 00. Thus, in this example, the bus/target/LUN information is `1/4/00`.

If your boot device is not the default boot device, enter the `show device` command and identify it from the list.

3. Boot the system from the installation CD-ROM or the RIS server. For example, for the devices identified above, if you are using the CD-ROM, enter the following:

```
>>> boot dka500
```

If you are using the RIS server, enter the following:

```
>>> boot ewa0
```

4. Exit the installation. If you have a VGA graphics console, choose to exit the installation, or choose Shell Window from the File menu of the Installation and Configuration Welcome dialog box. If you have a serial console terminal, select option 3) `Exit Installation`.

After you have determined your system configuration, do the following:

1. If the root domain is mountable when you boot from the installation CD-ROM or RIS server, the installation procedure attempts to read the existing device database from the installed root domain. If the hardware database read fails, you will get an error message. If this happens, you must translate the UNIX device name assignments to the proper hardware device.

Use the `/sbin/hwmgr` command with the `-view devices` option to identify the device to be restored by its bus/target/LUN (as explained in the previous step), to locate disks that could be used for backup, and to verify that a tape device is installed.

```
# /sbin/hwmangr -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
46:/dev/ntape/tape0    DEC TLZ10 (C)         DEC
bus-1-targ-4-lun-0
```

According to the hardware database, in this example the disk with bus/target/LUN of 1/4/00 is identified as `dsk0`. Assume that `/dev/disk/dsk0a` is the volume containing the corrupted root domain.

To visually confirm that you have identified the correct device, use the `hwmangr` command with the `-flash` option to cause the disk's light to flash for thirty seconds.

```
# /sbin/hwmangr -flash light -dsf /dev/disk/dsk0a
```

2. If you need to install a tape device for backup at this time, enter the following:

```
# /sbin/dn_setup -install_tape
```

3. If you have a good backup, you can restore from it. If your backup is out of date, you can restore from it and then use the `salvage` command with the `-d` option to recover later information. If your backup is good, skip to step 9. If you need to salvage files to use in conjunction with your backup, skip to step 8.
4. Mount the old root domain:

```
# mount old_root_domain#root /var/mnt
```

If the mount fails, skip to step 7. You cannot perform a backup.

5. Create a backup if possible. See Chapter 4 for more information.
6. Run the `/sbin/advfs/verify` command with the `-a` option. See Section 6.2.4 for more information. If this is successful and your system shows no errors, skip to Step 12.
7. Run the `/sbin/advfs/fixfdmn` command. See Section 6.2.5 for more information. If this is successful, skip to Step 12.
8. Recover files with the `salvage` command and save them to a temporary fileset or tape. If you have an old backup, run the `salvage` command with the `-d` option. If you do not specify the option, you will salvage the entire domain.

For example, to recover files from a corrupted root domain on volume `/dev/disk/dsk0a` and save them on the local, unused disk, `/dev/disk/dsk2c`, do the following:

- a. Create a temporary domain and filesets to hold the recovered information and mount the filesets.

```
# mkfdmn /dev/disk/dsk2c RECOVER
# mkfset RECOVER recover_fset
# mkdir /var/recover
# mount RECOVER#recover_fset /var/recover
```

- b. Run the `salvage` command. You must use the `-V` option to specify the volume that the command operates on. If you are salvaging files to be used with an older backup, use the `-d` option to salvage only the information that has changed since the date of the backup.

In this example, files from the root domain are extracted and stored in filesets mounted at `/recover`.

```
# /sbin/advfs/salvage -D /var/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-Dec-2001
salvage: Loading file names for all filesets: 09-Dec-2001
salvage: Starting recovery of all filesets: 09-Dec-2001
```

- c. View the `salvage.log` file to ensure that all necessary files were recovered. Once you have run the `mkfdmn` command in the next step, you cannot rerun the `salvage` command to extract more information.

9. 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
```

10. If you have a good backup source, restore the files from it.

```
# vrestore -xf /dev/tape/tape0 -D /var/mnt
```

11. If you have salvaged files, copy them from the recovery location to the root domain and remove the recovery domain.

```
# cd /
# cp -Rp /recover/root/* /var/mnt
```

```
# umount /mnt /recover
# rmfdmn RECOVER
rmfdmn: remove domain RECOVER [y/n] y
rmfdmn: domain RECOVER removed.
```

The root domain is restored.

12. Halt and reboot the system.

If the procedure was not successful, you will have to reinstall the operating system and recreate your files.

6.3.5 Recovering from Hardware Corruption

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 the AdvFS file system.

If unexplained errors occur on a volume in a multivolume domain, do one or more of the following:

- Examine the `/var/adm/messages` file for AdvFS I/O error messages. You must be root user to look at this file. The file 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. The file may look similar to the following:

```
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
```

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 `/sbin/advfs/verify` utility (Section 6.2.4) or the `/sbin/advfs/fixfdmn` command with the `-n` option (Section 6.2.5) to check the consistency of the domain's metadata.

- Check for device driver error messages for the volume described in the AdvFS I/O error message. 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. Unmount the domain as soon as possible and run the `verify` utility or the `fixfdmn` command with the `-n` option to check the integrity of the domain's metadata.
- If you have a multivolume domain and enough room, try to remove the faulty volume by using the `rmvol` utility as described in Section 2.3.5. If this succeeds, the file system problems should not recur.

If `rmvol` fails, you must recreate the domain.

1. If you have a recent backup, you can use it to restore the domain. If you have no backup, or if the backup is too old, use the `salvage` utility (Section 6.2.6) to extract the contents of the corrupted domain.
2. Remove the faulty domain by using the `rmfdmn` command.
3. Recreate the domain by using the `mkfdmn` command. The new domain has a domain version number (DVN) of 4 by default. (See Section 2.3.3 for more information.) Add volumes as needed. Do not to include the faulty volume in the new domain.
4. Restore the contents of the recreated domain using the salvaged and backup information.
5. Remount the filesets in the domain.

6.3.6 Moving an AdvFS Disk to Another Machine

If a machine has failed, you can move disks containing AdvFS domains to another computer running the AdvFS software. Connect the disks to the new machine and modify the `/etc/fdmns` directory so the new system recognizes the transferred volumes. You might need to reassign the disk SCSI IDs to avoid conflicts with the new system. (See your disk manufacturer instructions for more information.) You must be root user to perform this operation.

You cannot move DVN4 domains to systems running Version 4 of the operating system software. Doing so generates an error message. (See Section 6.4.6 for a discussion of compatibility.) 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. Executing the command

deletes all data on the disk you are moving. See Section 6.2.6 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 `/sbin/advfs/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.

In the following example, the system has a domain, `testing_domain`, on two disks on the old machine called `dsk3` and `dsk4`. This domain contains two filesets: `sample1` and `sample2`. These filesets are mounted on `/sample1` and `/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 in single-user mode.
4. Determine the device nodes created for the new disks.

```
# /sbin/hwmggr -show scsi -full
```

Check the Device Name column to find the names for the disks that you just added. These have the highest disk numbers.

5. Assume the new disk IDs in this example are `disk 7` and `disk 8`. Modify your `/etc/fdmns` directory to include symbolic links from the transferred domains.

```
# mkdir -p /etc/fdmns/testing_domain
# cd /etc/fdmns/testing_domain
# ln -s /dev/disk/dsk7c .
# ln -s /dev/disk/dsk8a .
# ln -s /dev/disk/dsk8b .
# ln -s /dev/disk/dsk8g .
# mkdir /sample1
# mkdir /sample2
```

6. Edit the `/etc/fstab` file to add the fileset mount-point information for the transferred filesets.

```
testing_domain#sample1 /sample1 advfs rw 1 0
testing_domain#sample2 /sample2 advfs rw 1 0
```

7. Mount the volumes.

```
# mount /sample1
# mount /sample2
```

6.3.7 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 fileset `usr` in `usr_domain`. The domain consists of the `dsk1g`, `dsk2c`, and `dsk3c` volumes. A backup tape of `/usr` and `/var` exists. (If you need to create a backup tape, see Chapter 4.) The procedure assumes that the root file system has already been restored. If it has not, see Section 6.3.4.

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 and restore from it.

```
# 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 additional 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. A backup tape of `/usr` and `/var` exists. (If you need to create a backup tape, see Chapter 4.) 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 6.3.4.

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 and restore from it.

```
# 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. Do a full restore of `/var` backup.

```
# cd /var
# vrestore -xv
```

6.3.8 Recovering from Accidental Use of the `mkfdmn` or `addvol` Command

The `mkfdmn` and `addvol` commands initialize a volume for use by AdvFS. If the volume is already part of an AdvFS domain, using these commands damages the metadata for the domain that existed on that volume. If you accidentally use one of these commands on a previously initialized volume, you might be able to recover some files by running the `salvage` utility with the `-S` option. See Section 6.2.6.3 for more information.

The more commands you execute after a `mkfdmn` or `addvol` command, the more you alter the original volume content and the less information you can recover with the salvage operation. If you cannot get a reasonable amount of information from running the `salvage` command, your only option is to restore the data from backup.

6.4 Preventing Problems

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. It is not a good idea to recover your system by using an operating system other than the one that crashed.

If a system crash has left your data compromised, see Section 5.5 for ways to improve the method data is written to disk in order to prevent such problems in the future.

6.4.1 Checking Free Space and Disk Usage

You can look at the way space is allocated on a disk by file, fileset, or domain. Table 6–4 describes command-line commands that you can use to examine disk space usage.

Table 6–4: Disk Space Usage Commands

Command	Description
<code>df^{ab}</code>	Displays disk space usage by fileset. Available space for a fileset is limited by the fileset quota if it is set. When both soft and hard quota limits are set, the command calculates the disk space available using the smaller of the two limits. 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. In domains with multiple filesets and no fileset quotas, the total capacity of all filesets can be greater than 100%. The available space value used in the calculation is all the space in the domain because it is available to each fileset.
<code>du</code>	Displays information about block allocation for files. Use the <code>-a</code> option to display information for individual files.
<code>ls</code>	Displays the space used by files. The <code>-l</code> option shows the space spanned by a sparse file. The <code>-s</code> option shows the actual block usage and might be more useful for examining sparse files.
<code>showfdmn^b</code>	Displays the attributes and block usage for each volume in an active domain. For multivolume domains, additional volume information is displayed.
<code>showfile</code>	Displays block usage and volume information for a file or for the contents of a directory.
<code>showfsets^b</code>	Displays information about the filesets in a domain. Use the <code>-q</code> option to display fileset quota limits.
<code>vd^{fab}</code>	Reformats output from the <code>showfdmn</code> , <code>showfsets</code> , and <code>df</code> commands. The characteristics of these three commands apply to the <code>vd</code> command.

^a For more accurate results, run the `quotacheck` command first. It is particularly important to do this on a quiescent system. If you do not have user and group quotas enabled, set them temporarily.

^b Choose an option to display in the units you want.

The `df` and `showfsets` commands do not include the space used for metadata and quota files, so the value in the Used field may be smaller than that actually used by the domain. The `vd` command displays disk usage

that includes metadata and quota files. See Section 3.4.2.4 and the reference pages for more complete information.

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

For information about examining disk usage related to quotas, see Section 3.4.

6.4.2 Reusing Volumes

If you want to add storage from an existing domain (the `/etc/fdmns` directory entry exists) to another domain, you can remove the volume from the original domain by using the `rmvol` command and then add it to the other domain. AdvFS automatically updates the domain bookkeeping.

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. You can do this without the `disklabel` command by answering **yes** to the prompt on the `addvol` or `mkfdmn` command. All information that is on the disk or disk partition that you are adding is lost. If you want to move a disk to use it on another system, see Section 6.3.6.

6.4.3 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. If you have the AdvFS Utilities, you can use the `addvol` command to add a new disk then the `rmvol` command to remove the damaged one. See the event management information in the *System Administration* manual for more information about examining disk activity.

6.4.4 Controlling Disk 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. User and group quotas, which are similar to UFS quotas, limit the amount of space a user or group can allocate for a fileset. Fileset quotas, which are unique to AdvFS, restrain a fileset from using all of the available space in a domain. See Chapter 3 for complete information on quotas.

6.4.5 Quota and Grace Period Limits

If you are close to your quota limit, or if it appears that you cannot get to that limit, you may be experiencing limitations of the quota system. See Chapter 3 for complete information on quotas.

6.4.5.1 Exceeding Quota Limits

If you are working in an editor and realize that the information you need to save exceeds your quota limit, do not abort the editor or write the file because data might be lost. Instead, do one of the following:

- Remove enough files to make room for the edited file before you save it.
- Save the file to another fileset, such as `tmp`, until you can remove files or increase quota limits for the fileset. Then move the file to your fileset.

6.4.5.2 Saving Up to Quota Limits

Hard and soft quota limits for the number of files are non-inclusive. You can only create files if you remain below the limit. For example, if your hard limit is 1000 files, you can only create 999 files.

Hard and soft quota limits for the number of blocks are similarly non-inclusive. In the rare case that you are 8K bytes below the user, group, or fileset quota and attempt to use some or all of the remaining space, you may be restricted by the quota limit. This is because AdvFS allocates storage in units of 8K bytes. If adding 8K bytes to a file meets or exceeds the quota limit, then that file is not extended.

6.4.5.3 Changing Quota Limits

If you set quota limits or grace periods for a fileset that is unmounted or does not have `userquota` and `groupquota` options in the `/etc/fstab` file (Section 2.4.1), the values you set are not retained. Update the `/etc/fstab` entry and mount your fileset before executing the `edquota` command.

6.4.5.4 Turning Off the Grace Period

To turn off a grace period, execute the `edquota` command to set the value to 1 second. Setting the value to 0 days imposes the default grace period of 7 days.

6.4.6 Avoiding Disk 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 2.3.3 for more information.) The newer operating system recognizes the older disk structure, but older operating systems do not recognize

the newer disk structure. If you install your Version 5 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 DVN of 3. 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. 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.

No tool exists to automatically upgrade DVN3 domains to DVN4. To upgrade a domain to DVN4, use the procedure in Section 2.3.3.4.

6.4.7 Avoiding Utility Incompatibility

Because Version 5.0 and later of the operating system have the new on-disk file formats, some AdvFS utilities from earlier releases might corrupt DVN4 domains. 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`

6.4.8 Avoiding Log File Inconsistency

When a system crashes, AdvFS performs recovery at reboot. Domains with filesets that were mounted at the time of the crash are recovered when one fileset in the domain is remounted. This recovery uses the AdvFS transaction log file to keep the AdvFS metadata consistent.

Since different versions of the operating system use different transaction log file structures, make sure 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 (Section 5.7) to promote a domain panic to a system panic, run the `/sbin/advfs/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, you can mount the filesets on a different version of the operating system, if appropriate.

6.4.9 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 increase the size of the volume containing the root domain or recreate the root domain on a larger volume.

6.4.9.1 Increasing the Size of the root Volume

If you increase the size of the underlying volume containing the root domain, you must notify the system of the change by executing the `mount` command with the `-o extend` option as described in Section 2.3.4.3. If your root volume is an LSM volume, see the *Logical Storage Manager* manual.

6.4.9.2 Replacing the root Volume with a Larger Device

To recreate the root domain on a different device, you need the new device installed, an installation CD-ROM or a RIS server, and a backup tape or an unused disk partition to back up the root domain.

To move your root domain to a new disk, you must first install the disk and have it recognized. If the disk that is to become the root volume is already installed on the system, skip to step 5.

1. Shut down the system.
2. Install the new disk device. For more information see your hardware manuals.
3. Verify that the console recognizes the newly added disk. In this example, DKB300 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. Reboot the system. During the boot process, the operating system recognizes the new device and updates the device information databases accordingly. Assume the operating system's device name for the added disk is `dsk6`.

To visually confirm that you have identified the correct device, use the `/sbin/hwmgrr` command with the `-flash` option to cause the disk's light to flash for thirty seconds.

```
# /sbin/hwmgrr -flash light -dsf /dev/disk/dsk6a
```

5. Log in as root and check the `/etc/fdmns` directory to determine the volume that contains root.

```
# ls -l /etc/fdmns/root_domain
total 0
lrwxrwxrwx  1 root      system 15 Feb 1 2002 dsk3a
-> /dev/disk/dsk3a
```

In this example `/dev/disk/dsk3a` is the volume containing the root domain.

6. If needed, use the `disklabel` command with the `-t advfs` option or the `diskconfig` utility (choosing AdvFS from the `Boot Block: list`) to configure the new disk as AdvFS bootable. You must be the root user to perform this operation.

```
# disklabel -r dsk6 > /tmp/backupdisklabel
# disklabel -R -r -t advfs dsk6 /tmp/backupdisklabel
# rm /tmp/backupdisklabel
```

7. Recreate the root domain on the new device.
 - a. Shut down the system booted from your old root domain.
 - b. Boot from the operating system CD-ROM or RIS server. For example, if you are using the CD-ROM:

```
>>> boot dka500
```

For example, if you are using the RIS server, enter the following:

```
>>> boot ewa0
```

- c. Exit the installation. If you have a VGA graphics console, choose to exit the installation, or choose Shell Window from the File menu of the Installation and Configuration Welcome dialog box. If you have a serial console terminal, select option 3) Exit Installation.

- d. Create a new root domain and root fileset on the new root device. Create a new entry in the `/etc/fdmns` directory to link to the original device.

```
# mkfdmn -r /dev/disk/dsk6a root_domain
# mkfset root_domain root
# mkdir /var/new_root
# mount root_domain#root /var/new_root
# mkdir /etc/fdmns/orig_root
# ln -s /dev/disk/dsk3a /etc/fdmns/orig_root
# mount orig_root#root /var/orig_root
```

- e. Dump the files from the old root domain and restore them to the new root domain.

```
# vdump -0f - /var/orig_root | vrestore -xf \
-D /var/new_root
```

8. 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 /var/new_root/etc/fdmns/root_domain
# ln -s /dev/disk/dsk6a dsk6a
# rm dsk3a
```

9. Halt the system and change the default boot device.

```
# halt
. . .
>>> set bootdef_dev dkb300
```

10. Boot the new root domain.

```
>>> boot
```

6.4.10 Memory Mapping, Direct I/O, and Data Logging Incompatibility

Unless you have turned on temporary atomic-write data logging by using the `mount` command with the `-o adl` option, 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 5.5, Section 5.6, and `mmap(2)`.

An file in memory-mapped mode is flushed by `smoothsync`, but may not be flushed according to `smoothsync_age`. This means that a file may remain temporarily in memory-mapped mode after being unmapped. See the *System Configuration and Tuning* manual for more information.

6.4.11 Preventing Access to Old Data

Obsolete data on disk might be recovered after a crash or when the `salvage` command is run unless old data is removed. To prevent access to this data, use the `chfsets` command with the `-o objectsafety` option. Choosing object safety prevents an application from reusing old data, but it does not guarantee complete data destruction.

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.

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

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

Object safety can degrade performance because pages must be written to disk twice.

6.4.12 Invalid or Corrupt Saveset Format

If you are restoring a saveset that has been saved on 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, so if you try to back to it, the saveset is not saved correctly.

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 4.3.7 for information about dumping to disk partitions.

6.4.13 Overcoming 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, system performance might degrade. If you have a multivolume domain, there are a number of ways that you can equalize the activity and increase throughput. See the *System Configuration and Tuning* manual and Chapter 5 for more complete information.

To discover the causes of poor performance, first check system activity. You can improve performance in a number of ways:

- Upgrade domains (Section 2.3.3.4)
- Eliminate disk access incompatibility (Section 5.6)
- Defragment domains (Section 5.9)
- Move filesets to different volumes (Section 5.14)

If you have AdvFS Utilities, you can also:

- Change resource allocation by increasing the size of a domain or substituting a faster volume for a slower one.
- Run the `vfast` utility (Section 5.8)
- Balance a multivolume domain (Section 5.11)
- Stripe individual files (Section 5.13)
- Migrate individual files (Section 5.12)

6.4.14 The `rmvol` or `migrate` Command Will Not Start

If you are removing a volume with the `rmvol` command (Section 2.3.5) and abort the command, on rare occasions the volume is left in an inaccessible state and you cannot write to it. These volumes are marked as “data unavailable” in the output of the `showfdmn` command. If this occurs, use the `chvol` command with the `-A` option to reactivate the volume.

A

Accessing File System Information with the SysMan Menu

You must be root user to use all the features of the SysMan Menu. You can open the menu by doing the following:

- From the command line enter:

```
# /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 icon.
 3. Select the SysMan Menu icon.

From the system management tasks list, to access AdvFS tasks, double click the following:

1. Storage
2. File System Management Utilities
3. Advanced File System (AdvFS) Utilities
4. The task you want to perform

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
<code>chfile</code>	Changes the attributes of a file
<code>chvol</code>	Changes the attributes of a volume
<code>defragment</code>	Makes the files in a domain more contiguous
<code>mkfdmn</code>	Creates a domain
<code>mkfset</code>	Creates a fileset within a domain
<code>renamefset</code>	Renames an existing fileset
<code>rmfdmn</code>	Removes a domain
<code>rmfset</code>	Removes a fileset from a domain
<code>/sbin/ad- vfs/switchlog</code>	Moves the AdvFS log file to a different volume in a domain

Table B–2: AdvFS Information Display Commands

Command	Description
<code>advfsstat</code>	Displays file system statistics
<code>ncheck</code>	Displays the tag and full pathname for each file in the file system
<code>showfdmn</code>	Displays the attributes of a domain
<code>showfile</code>	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
/sbin/advfs/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
/sbin/advfs/advscan	Locates AdvFS partitions on disks
/sbin/advfs/fixfdmn	Fixes on-disk metadata corruption
/sbin/ad- vfs/mountlist	Checks for mounted AdvFS filesets
/sbin/advfs/salvage	Recovers file data from damaged AdvFS domains
/sbin/ad- vfs/tag2name	Prints the pathname of a file given the tag number
/sbin/advfs/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
quotaon	Turns quotas on
repquota	Summarizes quotas for a file system

Table B–6: AdvFS Disk Structure Commands

Command	Description
/sbin/advfs/nvbmtpg	Displays a formatted page of the bitfile metadata table (BMT)
/sbin/ad- vfs/nvfragpg	Displays file fragment information
/sbin/advfs/nvlogpg	Displays a formatted page of the log
/sbin/advfs/nvtagpg	Displays a formatted page of the tag directory
/sbin/ad- vfs/savemeta	Captures a domain's metadata
/sbin/advfs/shblk	Displays unformatted disk blocks
/sbin/ad- vfs/shfragbf	Displays file fragment information
/sbin/advfs/vfilepg	Displays a page of an AdvFS file
/sbin/advfs/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
dmapi	Accesses the data management applications programming interface (DMAPI)
dtadvfs	Starts the AdvFS GUI
lsmsa	Starts Logical Storage Manager Storage Administrator (LSMSA)
migrate	Moves a file to another volume in the domain
mktrashcan	Attaches directories to a trashcan directory, which stores deleted files
rmtrashcan	Detaches a specified directory from a trashcan directory

Table B-7: AdvFS Utilities Commands (cont.)

Command	Description
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
vfast	Reduces file fragmentation and balances volume free space and I/O load

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. A one-to-one correspondence does not exist 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 <code>iostat</code> or system monitoring tools such as <code>sys_check</code> 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 <code>cp</code> and <code>mv</code> .
mkfdmn, mkfset	Use <code>newfs</code> .
mktrashcan, rmtrashcan, shtrashcan	Use CDE windowing environment, which provides a deferred deletion.

Table B-8: Comparison of AdvFS and UFS Commands (cont.)

AdvFS Command	UFS Command
mountlist	Use mount.
ncheck	Same as AdvFS.
fixfdmn, nvbmtpg, 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 <i>/etc/fstab</i> 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 of the operating system software. If you upgrade your system, recreate the domains, and restore the data from backup (Chapter 4), your AdvFS filesets and domains have the updated structure. (See Section 2.3.3.)

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:
5. Reboot the system in single-user mode. See the *System Administration* manual 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` command with the `-F` option 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 the *System Administration* manual 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` command with the `-F` option 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

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 there is a device called `disk_name`.

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_name
```

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 File System to AdvFS

By converting your 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 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 UFS file system to the equivalent 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` command with the `-F` option 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 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 `/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` command with the `-F` option 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 aUFS file system on one disk to the equivalent 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 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_name`):

```
# newfs /dev/disk/dsk2a disk_name
```
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/dsk2label  
# disklabel -t ufs -r -R /dev/rdisk/dsk2a \  
/tmp/dsk2label <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:

```
# rmdmn 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

Data Management Applications Programming Interface (DMAPI)

DMAPI is a Common Applications Environment (CAE) standard that allows data management applications to interface with the underlying operating system. An example of a data management application is a hierarchical storage manager (HSM), which can use DMAPI features to migrate data to and from tertiary storage as well as to track file information.

DMAPI functionality requires an AdvFS Utilities license. See `dmapi(3)` to identify the data management operations that are supported.

Use the `showfsets` command to determine if a file is enabled for DMAPI. To enable a fileset for DMAPI, execute the `chfsets` command. You must have root user privilege.

A number of restrictions apply to DMAPI-enabled filesets:

- The DMAPI-enabled fileset must be the only fileset in its domain. You cannot clone or add a fileset to a domain that contains a DMAPI-enabled fileset.
- Filesets that are named `root`, `usr`, and `var` cannot be enabled for DMAPI. Also, DMAPI filesets should not be mounted on `/`, `/usr`, and `/var`.
- Only DVN4 domains are supported under DMAPI.
- Only one application should manage the DMAPI-enabled fileset. Allowing more than one application to operate on a fileset is not supported.
- Filesets must be unmounted before you can turn on DMAPI functionality.
- Do not mount a DMAPI-enabled fileset on a version of Tru64 UNIX that does not support DMAPI. This could result in file corruption.
- Tru64 UNIX does not support importing a fileset from another system if either an existing or the imported fileset is enabled for DMAPI.
- You cannot dump a DMAPI-enabled fileset by using the `vdump` command. Use the vendor-approved backup for your data management application.
- Quotas are not supported on DMAPI-enabled filesets.
- Do not use the `salvage` command to restore lost data on a DMAPI-enabled fileset. The `salvage` utility operates only on primary

storage. This means that if the command is used on a DMAPI-enabled fileset, some files might be missing data that was made nonresident by the DMAPI application. Follow the procedures provided by the DMAPI application vendor.

- Utilities that manage storage, such as `defragment`, `balance`, `migrate`, and `/sbin/advfs/verify`, work on DMAPI-enabled filesets. However, running these utilities might result in less effective storage management than would occur without DMAPI being enabled.
- In the current DMAPI implementation, asynchronous event posting is reliable in all but the most unusual cases. For more information, see the CAE DMAPI standard under Event Types.
- Memory mapping is not supported on DMAPI-enabled filesets.
- The `vfilepg` command exits if a page is sparse. Because a page migrated out to tertiary storage by DMAPI appears as sparse in primary storage, the `vfilepg` command may not operate properly on DMAPI-enabled filesets.

To create a new domain and fileset and enable DMAPI for the fileset:

1. Create a new domain and fileset.
2. Turn on DMAPI functionality by using the `chfsets` command with the `-o dmapi` option.
3. Check the fileset status by using the `showfsets` command.

For example, to create the fileset `sales_fset` in the domain `receipt_dmn`:

```
# mkfdmn /dev/disk/dsk1c receipt_dmn
# mkfset receipt_dmn sales_fset
# chfsets -o dmapi receipt_dmn sales_fset
# showfsets receipt_dmn
```

To enable an existing fileset for DMAPI:

1. Be certain only one fileset exists in the domain. If it does not:
 - a. Dump the fileset by using the `vdump` command.
 - b. Create a new domain and a new fileset. Because it is in a new domain, the fileset can have the same name as the old fileset.
 - c. Restore the fileset to the new domain by using the `vrestore` command.
2. Unmount the fileset if needed.
3. Enable DMAPI by using the `chfsets` command.
4. Check the fileset status by using the `showfsets` command.

To disable DMAPI for a fileset:

1. Unmount the filesets.
2. Call the `chfsets` command with the `-o nodmapi` option.
3. Check the fileset status by using the `showfsets` command.

Note that you can disable DMAPI only if the data management application has not begun modifying the fileset. In order to avoid the possibility of corruption, if it has begun you cannot disable DMAPI.

To check if DMAPI-managed regions are enabled on a file, execute the `showfile` command.

```
# showfile /mnt/file1
  Id Vol PgSZ Pages XtntType Segs SegSZ   I/O Perf File
6.8006  1  16    4  simple  **   **  async 100% file1
DMAPI regions present on this file.
```


E

Graphical User Interfaces

Two interfaces provide graphical representation of the AdvFS file system: the Logical Storage Manager Storage Administrator (LSMSA) and the AdvFS Graphical User Interface (AdvFS GUI). With LSMSA you can manage LSM objects and perform LSM as well as the AdvFS file system administrative functions. The AdvFS GUI provides a visual representation of the AdvFS file system. Both require utilities licenses. For information about license activation contact your software support organization.

LSMSA is a Java-based application that operates on local or remote (client) systems. LSMSA operations are described in detail in the *Logical Storage Manager* manual.

AdvFS GUI operations are described in this appendix. 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.

If you are running a cluster configuration, the AdvFS GUI accesses it through a local node. You can use the graphical interface to manage the complete cluster, but the AdvFS GUI is not optimized for cluster management.

E.1 Installing the AdvFS GUI

To load the AdvFS GUI, choose the AdvFS Utilities subset when you update or install the operating system.

Table E-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 E-2.

Table E-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 E-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 E-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

E.2 Components of the GUI

The GUI has two parts: `advfsd`, the agent, and `dtadvfs`, the actual graphical interface. The agent must be running for the GUI to operate.

E.2.1 GUI Agent (advfsd)

The AdvFS GUI **agent**, `advfsd`, executes 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)`.) You can stop the agent and restart it when you run the GUI. This is useful if you think the agent is slowing your system.

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 execute 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 E.3.1. The interval is saved between restarts of the agent.

E.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 E.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 E.3.1. The GUI refresh interval is not saved between restarts of the GUI.

E.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. It contains 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 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 must also know that password.

See `advfsd(8)` and `dtadvfs(8)` for more information.

E.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.

E.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.

E.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.

E.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.

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.

E.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.
- If you increase the size of a LSM volume that is in use by an AdvFS domain, you cannot use the AdvFS GUI to notify the system of the change. You must use the `mount` command with the `-o extend` option on the command line. (See Section 2.3.4.3 and `mount(8)`.)
- 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.

E.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:

- 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.

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.

E.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.

E.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.

E.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.

E.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.

E.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.

E.4.1.1 Creating a Domain

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, do the following:

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.

E.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.

E.4.1.3 Enlarging a Domain

Add volumes to transform a single-volume domain (except the root domain which, unless it is in a cluster configuration, can only have one volume) into a multivolume domain or to enlarge a multivolume domain. (See Section 2.3.4.)

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 have 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.

E.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 to delete all filesets, its entry in the `/etc/fdmns` directory is removed. However, this can present a security hole because only pointers are changed and no data on the volume is removed. It might be possible to access the fileset data from the command line by using the `salvage` utility. (See Section 6.2.6.) If you need the increased security, remove each fileset individually.

After you delete a domain, you no longer have access to its filesets. 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 get error messages when you reboot the system or when you enter a `mount` command with the `-a` option 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.

E.4.1.5 Renaming a Domain

You can assign a new name to an existing domain. (See Section 2.3.7.) 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` command with the `-a` option 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.

E.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. Make sure that you choose to modify 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.

E.4.2.1 Creating a Fileset

You cannot create a fileset until you have a domain with which to associate it. (See Section E.4.1.1.)

To create a fileset:

1. In the object tree, highlight the domain for which the fileset is to 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 (if wanted).

When you create a fileset, you can set the free space alert only as a percentage of the available space.

E.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.

E.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.

E.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.

E.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.

E.4.2.6 Removing a Fileset

You must unmount a fileset before you can delete it. You can unmount it 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.

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 6.2.6.) If you are concerned about security, remove each fileset individually.

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.

E.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.

E.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.

E.4.3.1 Creating an AdvFS Fileset Clone

Create a fileset clone to back up files while the file system is on line. (See Section 4.1.2.1.) 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.

E.4.3.2 Mounting a Clone

You must mount a clone 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.

E.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.

E.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.

E.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.)

E.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.

E.4.4.2 Adding a Volume

Adding a volume is the same as enlarging a domain. (See Section E.4.1.3.) See Section 2.2 for detailed information about volumes.

E.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 2.3.5.) The data that was stored on that volume is moved to other volumes in the domain. 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.

E.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.

E.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 5.11.) You cannot balance a domain while you are running the `vfast` command with the `-o balance=` option, defragmenting, adding or removing volumes, or removing filesets on the same domain. 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.

You can choose to abort the operation at any time. The files that were defragmented remain defragmented.

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.

E.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 5.9.) You cannot defragment a domain while you are running the `vfast` command with the `-o defragment=` option, 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.

E.5 GUI Operation

Table E-3 presents some GUI problems and solutions. Check `dtadvfs(8)` and `advfsd(8)` for additional information.

Table E-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 <code>dtadvfs</code> from the command line	You are not the root user.
Password Error dialog when starting <code>dtadvfs</code> 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 <code>disks.ignore</code> file. If none of these 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.

Table E-3: GUI Troubleshooting (cont.)

Problem	Cause/Solution
Domain deleted but associated volumes shown as in use	Manually change the disk label on the device; use the command line <code>disklabel</code> command or the CDE Disk Configuration utility.
Volume removed but shown	Manually change the disk label on the device; use the command line <code>disklabel</code> command or the CDE Disk Configuration utility.
<code>advfsd</code> consuming high CPU and I/O resources	Change the agent state monitor interval to reduce the polling frequency. (See Section E.2.1). Decrease the number of LSM disk groups and volumes. Decrease the number of unmounted filesets. Inactive domains are particularly slow. Use the <code>disks.ignore</code> file. (See Section E.2.2.2). If none of these actions solve 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 <code>/etc/fstab</code> file. Do one of the following in single-user mode: Repair the domain or move the bad domain from the <code>/etc/fdmns</code> directory to another directory. (The latter makes repair difficult.) Reboot to multiuser
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 E.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.

F

Application Programming Interface (API)

Use the API to build customized backup and restore capabilities. The API consists of base system functions, that are described in more detail in the reference pages.

- `advfs_clonelfset(3)`
Creates a read-only copy of an active fileset
- `advfs_get_fdmn_list(3)`
Gets a domain list
- `advfs_get_file_attributes(3)`
Gets the attributes of an AdvFS file
- `advfs_get_fset_list(3)`
Obtains a list of all the filesets in a domain
- `advfs_get_fset_quotas(3)`
Obtains fileset quotas for a fileset
- `advfs_rmfset(3)`
Removes a fileset or a fileset clone
- `advfs_set_file_attributes(3)`
Sets AdvFS file attributes
- `advfs_set_fset_quotas(3)`
Set fileset quotas for a fileset

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 executes 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 8K bytes) 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

An AdvFS fileset clone is a read-only snapshot of fileset data at the time the clone was created. It is updated as the fileset is updated. AdvFS fileset cloning is analogous to the HSG80 controller "ADD SNAPSHOT_UNITS" command.

An HSG80 controller clone is a complete physical copy of the system so there is no need to track changes.

A system clone is a complete copy of an operating system used to generate the system on more than one installation.

See also 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.

In networking terminology, a domain is a collection of related systems on a subnetwork of an intranet or the Internet.

In Windows NT, a domain is a collection of computers defined by the administrator of a Windows NT Server network that share a common directory database.

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 8K bytes.

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 8K bytes (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 file characteristics.

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 8K bytes of contiguous disk space (16 blocks).

PAK

Product authorization key. License to access Hewlett-Packard Company 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.

An LSM volume is an object that represents a hierarchy of mirrors, subdisks, and LSM disks in a disk group. Applications and file systems make read and write requests to the LSM 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

A

- adding a volume**, 2–8
- adding volumes**, E–16
- addvol command**, 2–8
 - used accidentally, 6–26
- AdvFS GUI agent**, E–2
- AdvFS Utilities**
 - commands, B–3
 - restoring license, 6–24
- AdvFS Utilities license**
 - restoring AdvFS Utilities license, 6–24
- AdvfsAccessMaxPercent attribute**, 5–12
- advfsd command**, E–2
- AdvfsDomainPanicLevel attribute**, 5–27
- AdvfsReadyQLim attribute**, 5–12
- advfsstat command**, 5–2
- AdvfsSyncMmapPages attribute**, 5–12
- advscan command**, 6–13
- allocation of storage**, 1–8
- API**
 - data management, D–1
- API for AdvFS**, F–1
- asynchronous I/O**, 5–7
- atomic-write data logging**, 6–33
 - and memory mapping, direct I/O, 6–33
 - asynchronous, 5–9
 - description, 5–8
 - persistent, 5–10
 - synchronous, 5–9
 - temporary, 5–10

- turning on and off, 5–11

- attributes**
 - changing, 5–12

B

- backup**
 - compressing, 4–16
 - creating source, 4–4
 - data integrity, 4–1
 - error protection, 4–16
 - fileset, 4–1
 - from mounted fileset, 4–4
 - incremental, 4–14
 - level, 4–14
 - methods of, 4–12
 - remote, 4–19
 - subdirectories, 4–15
 - to files, 4–15
 - to standard output, 4–15
 - to tape, 4–14
 - using controller-based clone, 4–8
 - using controller-based snapshot, 4–10
 - using fileset clone, 4–5
 - using LSM split mirror, 4–6
 - with trashcan, 2–27
- balance command**, 5–22
- balancing domains**, 5–22, E–17
- bitfile metadata table**
 - (See BMT)
- bitmap**
 - storage, 1–7
- BMT**, 1–7

C

- cache tuning**, 5–6
- chfile command**, 4–2, 5–8
- chfsets command**, 2–21, 3–14, 3–16, 6–34
- chvol command**, 2–2, 5–12, 6–35
- clone, fileset**
 - backing up, 4–5
 - creating, 2–23, E–14
 - deleting, 2–24, E–15
 - mounting, 2–24, E–15
 - removing, 2–24, E–15
 - renaming, 2–24
 - unmounting, 2–24, E–15
- clonefs command**, 2–23
- Collect**, 5–2
- commands**
 - AdvFS base, B–1
 - AdvFS Utilities, B–3
 - UFS comparison, B–4
- compressing savesets**, 4–16
- configuring**
 - AdvFS root, 2–24
 - file system, 1–6
- converting**
 - AdvFS to UFS
 - fileset, C–14
 - root, C–13
 - UFS to AdvFS
 - data, C–7
 - root file system, C–5
 - /usr file system, C–1
- copy-on-write**, 2–22
- corruption**
 - due to old utilities, 6–30
 - fixing metadata, 6–4
 - fixing root domain, 6–17
 - hardware, 6–21
- crash recovery**, 6–1
- creating**
 - domain, 2–4, E–8
 - fileset, 2–18, E–11
 - fileset clone, 2–23, E–14

D

- data**
 - logging, 5–7, 6–33
 - management API, D–1
 - restoring remotely, 4–19
 - restoring with vrestore, 4–19
 - reusing, 6–34
 - security, 6–34
- data cache tuning**, 5–6
- Data Management API**, D–1
- defragment command**, 5–18
- defragmenting**, 5–18, E–17
 - file, 5–22
 - with vfast, 5–13
- deleting**
 - domain, 2–14, E–10
 - fileset, 2–20, E–13
 - fileset clone, 2–24, E–15
 - fileset quotas, 3–16
 - user and group quotas, 3–14
- device**
 - (*See* volume)
- devices and volumes window**, E–8
- df command**, 3–21, 6–27
- direct I/O**, 5–11, 6–33
- disk**, 2–1
 - (*See also* volume)
 - bad usage information, 6–27
 - damaged, 6–22
 - dumping to a partition, 4–15
 - failure recovery, 6–21
 - moving, 6–22
 - removing, 6–28
 - space used, 3–17, 6–27, 6–28
 - version incompatibility, 6–29
- disks.ignore file**, E–4
- displaying**
 - domain information, 2–3
 - file information, 5–23
 - space used, 3–17, 6–27
- DMAPI**, D–1
- domain**
 - adding a volume, 2–8

- adding volumes, E-9
- balancing, 5-22, E-17
- corrupted root, 6-17
- creating, 2-4, E-8
- creating DVN3, 2-7
- creating DVN4, 2-6
- damaged, 6-4
- defragmenting, 5-18, E-17
- deleting, 2-14, E-10
- description, 1-3
- disks of, 6-13
- enlarging, 2-8, E-9
- incompatibility, 6-29
- naming, 2-2
- panic, 6-8
- quiescing, 4-3
- recovering, 6-1
- recovering file data, 6-5
- recovering multivolume, 6-24
- reducing size, 2-12
- removing, 2-14, E-10
- removing volumes, 2-12
- renaming, 2-14, E-11
- renaming root, 2-25
- repairing, 6-13
- root, 2-24
- security, 2-14, E-10
- setting free space alert, E-9
- space usage, 6-27
- striping, 5-25
- upgrading to DVN4, 2-7
- Version 5, 2-5
- domain panic**, 6-8
- domain version number**, 2-5
- dtadvfs command**, E-3
- du command**, 6-27
- dumping**
 - error protection, 4-16
 - incremental, 4-14
 - remote, 4-19
 - subdirectories, 4-15

- to a disk partition, 4-15
- to a file, 4-15
- to standard output, 4-15
- to tape, 4-14

DVN

(See domain version number)

E

edquota command, 3-5, 3-10

enlarging

- domain, 2-8, E-9
- root, 6-31

/etc/fdmns directory

- description, 2-3
- restoring, 6-12

/etc/fstab file

- designating filesets, 2-16
- quotas, 3-4, 3-12

extent, 1-4

F

fdmns directory, 2-3

file

- defragmenting, 5-22
- defragmenting with vfast, 5-13
- deleting with trashcan, 2-27
- disks.ignore, E-4
- extent, 1-4
- frag, 1-4, 5-2
- GUI log, E-4
- gui.passwd, E-3
- hosts.allow, E-4
- moving, 5-26
- preallocation, 1-4
- quota, 3-4, 3-14
- recovering, 6-1
- restoring remotely, 4-19
- restoring with vrestore, 4-20
- sparse, 1-4

- storage allocation, 1–4
- striping, 5–25
- file quotas**
 - (*See* quotas)
- file system**
 - AdvFS root, 2–24
 - configuring, 1–6
 - converting AdvFS to UFS, C–13
 - converting UFS to AdvFS
 - file system, C–7
 - root, C–5
 - /usr, C–1
 - design, 1–2
 - monitoring remotely with GUI, E–5
 - recovering, 6–1
 - verifying, 6–2
 - viewing with GUI, E–6
- fileset**
 - backing up, 4–1
 - backup using fileset clone, 4–5
 - changing characteristics, 2–21
 - cloning
 - (*See* clone, fileset)
 - converting AdvFS to UFS, C–14
 - creating, 2–18, E–11
 - deactivating quotas, 3–16
 - deleting, 2–20, E–13
 - description, 1–3
 - disk usage, 6–27
 - displaying information, 2–17
 - grace period, 3–14
 - mounted read-only, 6–10
 - mounting, 2–19, E–12
 - moving, 5–26
 - naming conventions, 2–15
 - object safety, 6–34
 - quotas, 3–14, E–13
 - read-only recovery, 6–10
 - removing, 2–20, E–13
 - renaming, 2–21, E–14
 - renaming root, 2–25
 - security, 2–20, 6–34, E–13
 - setting free space alert, E–12
 - space usage, 6–27

- unmounting, 2–20, E–12
- upgrading, 2–18
- fileset clone**
 - (*See* clone, fileset)
- fixfdmn command**, 6–4
- frag file**, 5–2
- fragments**, 1–4, 5–2
- free space alert**, E–7
 - domain, E–9
 - fileset, E–12
 - volume, E–16
- freezefs command**, 4–3
- fstab file**, 2–16

G

- grace period**
 - definition, 3–3
 - setting
 - fileset, 3–16
 - user and group, 3–5
 - turning off, 3–3
- graphical user interface**
 - (*See* GUI)
- group quotas**, 3–4
- growing a domain**, E–9
- GUI**
 - agent, E–2
 - allowed hosts, E–4
 - and LSM volumes, E–5
 - devices and volumes window, E–8
 - display units, E–7
 - domain
 - adding volumes, E–9
 - balancing, E–17
 - creating, E–8
 - defragmenting, E–17
 - deleting, E–10
 - enlarging, E–9
 - free space alert, E–9
 - removing, E–10
 - renaming, E–11
 - fileset
 - creating, E–11

- deleting, E-13
- free space alert, E-12
- mounting, E-12
- removing, E-13
- renaming, E-14
- setting quotas, E-13
- unmounting, E-12
- fileset clone
 - creating, E-14
 - deleting, E-15
 - mounting, E-15
 - removing, E-15
 - unmounting, E-15
- ignoring disks, E-4
- installation, E-1
- log files, E-4
- main window, E-7
- monitoring remote file systems,
E-5
- password file, E-3
- security, E-3
- software subsets, E-1
- troubleshooting, E-18
- volume
 - adding, E-16
 - removing, E-16
 - setting free space alert, E-16

gui.passwd file, E-3

H

hard limit, 3-2
hosts.allow file, E-4

I

I/O

- asynchronous, 5-7
- atomic-write, 5-8, 6-33
- direct, 5-11, 6-33
- synchronous, 5-7

installation

- AdvFS GUI, E-1
- upgrade, 6-29

iostat command, 5-2

L

license

- restoring, 6-24

limits

- exceeding quota, 6-29
- quota, 3-2

log

- transaction
(See transaction log file)

logging, 5-8

ls command, 3-4, 6-27

LSM Storage Administrator, E-1

LSMSA, E-1

M

memory mapping, 6-33

metadata

- backing up, 4-1
- consistency, 4-3
- description, 1-7
- fixing corruption, 6-4
- saving, 6-2

migrate command, 5-23

migrating files, 5-23

mkfdmn command, 2-4

- used accidentally, 6-26

mkfset command, 2-18

mktrashcan command, 2-27

mount command, 2-19

mounting

- fileset, 2-19, E-12
- fileset clone, 2-24, E-15

mountlist command, B-2

moving

- between AdvFS and UFS, C-1

disk, 6–22
files, 5–26
filesets, 5–26

N

naming

domains, 2–2
filesets, 2–15

ncheck command, 3–17

nvbmtpg command, B–3

nvfragpg command, B–3

nvlogpg command, B–3

nvtagpg command, B–3

O

object safety, 6–34

P

page

zero-fill, 2–21

panic

domain, 6–8

partition

finding AdvFS, 6–13

reusing, 6–28

performance

improving, 6–34

improving by changing attributes,
5–12

improving by disabling frag file,
5–2

improving transaction log file, 5–3

monitoring, 5–2

with vfast, 5–13

Q

quiescing a domain, 4–3

quot command, 3–17

quota command, 3–18

quotacheck command, 3–19

quotaoff command, 3–13, 3–14

quotaon command, 3–4, 3–5, 3–13

quotas, 3–1

activating fileset, 3–16

activating user and group

at start-up, 3–12

manually, 3–13

changing GUI view, E–7

checking disk usage, 3–17

checking fileset ownership, 3–17

comparing types, 3–4

deactivating fileset, 3–16

deactivating user and group, 3–14

deleting fileset, 3–16

deleting user and group, 3–14

disabling fileset, 3–16

disabling user and group, 3–14

displaying disk usage, 3–18

enabling fileset, 3–16

enabling user and group

at start-up, 3–12

manually, 3–13

/etc/fstab file, 2–16, 3–4, 3–12

exceeding limits, 6–29

file, 3–14

file size, 3–4

fileset, 3–14, E–7, E–13

grace period, 3–3

group, 3–4

hard limit, 3–2

limits, 3–2

multiple filesets, 3–16

multiple users and groups, 3–10

relocating files, 2–16

removing fileset, 3–16

removing user and group, 3–14

restoring, 4–20

setting

fileset, 3–14, E–13

multiple filesets, 3–16

multiple users and groups, 3–10

user and group, 3–5

- soft limit, 3-2
- starting fileset, 3-16
- starting user and group
 - at start-up, 3-12
 - manually, 3-13
- summarizing by fileset, 3-19
- turning off fileset, 3-16
- turning off user and group, 3-14
- user and group, 3-5
- users, 3-4
- viewing with GUI, E-7

R

read-only fileset, 6-10

recovering

- domain, 6-1
- file data, 6-5
- files from trashcan, 2-27
- fileset mounted read-only, 6-10
- from mkfdmn or addvol, 6-26
- multivolume domain, 6-24
- root domain, 6-17
- system crash, 6-1

refresh interval, E-3

removing

- bad disk, 6-28
- domain, 2-14, E-10
- fileset, 2-20, E-13
- fileset clone, 2-24, E-15
- fileset quotas, 3-16
- user and group quotas, 3-14
- volume, 2-12, E-16

renamefileset command, 2-21

renaming

- domain, 2-14, E-11
- fileset, 2-21, E-14
- fileset clone, 2-24

repairing

- domain, 6-13

repquota command, 3-19

restoring

- AdvFS Utilities license, 6-24
- data, 4-19
- /etc/fdmns directory, 6-12
- files with vrestore, 4-20
- quotas, 4-20
- remote, 4-19
- saveset, 4-20

rmfdmn command, 2-14

rmfset command, 2-20

rmtrashcan command, 2-27

rmvol command, 2-12

root

- converting AdvFS to UFS, C-13
- domain, 2-24
- file system, 2-24
- fixing corruption, 6-17
- increasing size, 6-31
- mounting in single-user mode, 2-25
- recovering, 6-17
- renaming, 2-25
- salvaging data from, 6-7

rvdump command, 4-19

rvrestore command, 4-19

S

salvage command, 6-5

savemeta command, 6-2

saveset

- bad, 6-34
- compressing, 4-16
- restoring, 4-20

security

- data, 6-34
- domain, 2-14, 6-34, E-10
- fileset, 2-20, E-13

shblk command, B-3

shfragbf command, B-3

showfdmn command, 3-22, 6-27

showfile command, 1-6, 5-23,
6-27

showfsets command, 2-17, 3-22, 6-27
shtrashcan command, 2-27
size
 increase root, 6-31
 quota files, 3-4
 space usage, 6-27
smoothsync_age attribute, 5-12
soft limit, 3-2
space
 corrupt disk usage output, 6-27
 disk usage, 6-27
sparse file, 1-4
storage
 bitmap, 1-7
 choosing, 1-8
stripe command, 5-25
striping files, 5-25
switchlog command, 5-3
synchronous I/O, 5-7
SysMan Menu, A-1

T

tag, 3-17
tag2name command, B-2
thawfs command, 4-3
transaction log file
 data logging, 5-7
 description, 1-3
 improving performance, 5-3
 incompatibility, 6-30
 isolating, 5-3
trashcan, 2-27
tuning
 attributes, 5-12
 data cache, 5-6

U

UBC, 5-6
UFS
 AdvFS command comparison, B-4

 converting AdvFS to, C-13
 converting fileset to, C-14
umount command, 2-20
unmounting
 fileset, 2-20, E-12
 fileset clone, 2-24, E-15
upgrading
 fileset, 2-18
 installation, 6-30
usage
 domain, 3-17, 6-27
 file, 6-27
 fileset, 6-27
user quotas, 3-4
/usr file system
 converting UFS to AdvFS, C-1
utility incompatibility, 6-30

V

vdf command, 3-23, 6-27
vdump command, 4-12
verify command, 6-2
version number, domain, 2-5
vfast command, 5-13
vfilepg command, B-3
viewing
 domain, 2-3
 file, 5-23
volume, 2-1
 activating, 6-35
 adding, 2-8, E-16
 attributes, 2-2
 changing, 5-26
 deleting, 2-12, E-16
 description, 1-3
 exchanging, 2-11
 increasing size, 2-11
 removing, 2-12, E-16
 reusing, 6-28
 setting free space alert, E-16
vrestore command, 4-19
vsbmpg command, B-3

Z

zero-fill pages, 6–34