

# **AdvFS Command Line and Application Programming Interface**

## **External Reference Specification**

**Version 1.13**

**JA**

**CASL**

**Not Inspected/Date Inspected**



Building ZK3  
110 Spit Brook Road  
Nashua, NH 03062

Copyright (C) 2008 Hewlett-Packard Development Company, L.P.

# Preface

Version 1.4 of the AdvFS Command Line and API External Reference Specification is being made available to all partners in order to allow them to design and implement code meeting the specifications contained herein.

<b>1</b>	<b>INTRODUCTION .....</b>	<b>4</b>
1.1	ABSTRACT .....	4
1.2	AUDIENCE.....	4
1.3	TERMS AND DEFINITIONS.....	4
1.4	RELATED DOCUMENTS .....	4
1.5	ACKNOWLEDGEMENTS .....	4
<b>2</b>	<b>PRODUCT FEATURES AND USAGE.....</b>	<b>5</b>
2.1	OVERVIEW .....	5
2.2	USE OF THE EXISTING COMMAND INTERFACE .....	5
2.3	JOURNALING FILE SYSTEM .....	5
2.4	MULTIPLE DISK SUPPORT .....	5
2.5	FREEZING AND THAWING THE FILE SYSTEM .....	6
2.6	BACKUP UTILITIES.....	6
2.7	TRASHCANS .....	6
2.8	ON-DISK STRUCTURE VIEWER.....	6
2.9	FILE SYSTEM EXPANSION .....	6
2.10	SNAPSHOT SUPPORT .....	6
2.11	STORAGE THRESHOLDS .....	7
2.12	COMMAND SUMMARY .....	7
<b>3</b>	<b>EXTERNAL REFERENCE SPECIFICATIONS.....</b>	<b>9</b>
3.1	INTERNAL INTERFACES .....	9
3.2	EXTERNAL INTERFACES .....	9
3.3	INTERFACE TO PROGRAMS.....	9
3.3.1	<i>Data Structures</i> .....	9
3.3.2	<i>advfs ioctl</i> .....	10
3.4	INTERFACE TO USERS .....	12
3.4.1	<i>newfs_advfs</i> .....	12
3.4.2	<i>mkfs_advfs</i> .....	14
3.4.3	<i>mount_advfs</i> .....	16
3.4.4	<i>umount_advfs</i> .....	19
3.4.5	<i>quotacheck_advfs</i> .....	20
3.4.6	<i>extendfs_advfs</i> .....	21
3.4.7	<i>fsadm_advfs</i> .....	22
3.4.8	<i>fsck_advfs</i> .....	34
3.4.9	<i>ncheck_advfs</i> .....	37
3.4.10	<i>quot_advfs</i> .....	38
3.4.11	<i>edquota, repquota, quotaon/quotaoff, quota</i> .....	39
3.4.12	<i>ff_advfs</i> .....	40
3.4.13	<i>advfreeze/advithawfs</i> .....	41
3.4.14	<i>advstat</i> .....	44

3.4.15	<i>advtrashcan</i> .....	45
3.4.16	<i>advdump</i> .....	46
3.4.17	<i>advrestore</i> .....	49
3.4.18	<i>advdiskusg</i> .....	53
3.4.19	<i>adv Salvage</i> .....	54
3.4.20	<i>advvods</i> .....	57
3.4.21	<i>adv Savemeta</i> .....	61
<b>4</b>	<b>PRODUCT ENVIRONMENT</b> .....	<b>62</b>
4.1	ASSUMPTIONS .....	62
<b>5</b>	<b>ISSUES</b> .....	<b>63</b>
5.1	OPEN ISSUES .....	63
5.1.1	<i>fsadm (migrate) options</i> .....	63
5.1.2	<i>Property list API</i> .....	63
5.1.3	<i>Management API</i> .....	63
<b>APPENDIX A: MIGRATION FROM TRU64 TO HP-UX</b> .....		<b>64</b>

# 1 Introduction

## 1.1 Abstract

The goal of this document is to outline the HP-UX AdvFS command line and public API interface. The syntax and functional descriptions of the various AdvFS commands and API will be presented here.

## 1.2 Audience

The assumption is that the audience has general file system and HP-UX knowledge.

## 1.3 Terms and Definitions

<b>DEV_BSIZE</b>	Block size. Currently it is 1KB on HP-UX.
<b>extent</b>	A mapping from file system virtual range to the physical disk.
<b>fsname</b>	The name for a multi-volume file system (including a snapshot).
<b>freeze</b>	Stop any metadata activity.
<b>metadata</b>	Data on the disk used by the file system to manage user data.
<b>snapshot</b>	A moment-in-time copy of an existing AdvFS file system that is mountable.
<b>special</b>	The block special device.
<b>tag</b>	Unique file identifier for an AdvFS file system. It is similar to an HFS inode.
<b>thaw</b>	Resume normal operation.
<b>volume</b>	A volume is any mechanism that behaves like a UNIX block device, such as a disk or logical volume that has been configured with the Logical Volume Manager (LVM).

## 1.4 Related Documents

- `fs_wrapper(2)` HP-UX man page

This man page describes the configuration and binary files used by file system administration commands on HP-UX.

## 1.5 Acknowledgements

BN, SM, and BC.

## 2 Product Features and Usage

### 2.1 Overview

The Advanced File System (AdvFS) features rapid crash recovery, high performance, and a flexible structure that enables management of the file system while it is online.

### 2.2 Use of the Existing Command Interface

AdvFS on HP-UX will fit into the existing file system interface that the current HP-UX file systems use. This will help to provide a short learning period for new users of AdvFS.

### 2.3 Journaling File System

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. By using journaling techniques, AdvFS provides faster crash recovery than the traditional UNIX File System (HFS).

### 2.4 Multiple Disk Support

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

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

One benefit of the two-layer design is that AdvFS can support multiple disk volumes per file system. A volume can be added or removed without taking the file system offline. The user can add volumes immediately after creating a file system, or later add volumes when the file system requires additional space.

For multi-volume support, AdvFS must be able to identify a file system by something other than the disk volume(s) on which it resides. AdvFS introduces the concept of a file system name (fsname) to decouple the name of the file system from its underlying storage. To leverage the existing infrastructure within HPUX and to encourage the adoption of AdvFS as the preferred file system, file system names are optional. The user can create a file system without the fsname parameter, but all multi-volume related features will require an fsname to be specified. The user can add an fsname to an AdvFS file system that was not created with one at any time. Once an fsname is assigned, it must be used for all AdvFS operations (i.e. fsadm). In order to use the fsname with commands, the user will have to use this format: `/dev/advfs/<fsname>`.

AdvFS file system names are maintained via the `/dev/advfs` directory. Each file system has an entry in `/dev/advfs` and the directory name defines the file system name. The directory contains symbolic links to each of the block special devices that comprise the storage for the file system.

AdvFS offers the user several ways to optimize performance in an fsname file system. AdvFS allows the transaction log to be placed on (or isolated to) any volume in the file system. For example, the log can be placed on a fast disk all by itself for better performance. Additionally, AdvFS has a utility that will balance data across all of the volumes in the file system. Both optimizations are done with the fsadm [3.4.7] command.

## 2.5 Freezing and Thawing the File System

To allow coherent hardware snapshots in multiple volume file system configurations, file system metadata must be consistent across all volumes when the individual volumes are snapped or cloned. These configurations include both multi-volume AdvFS file systems and logical volumes.

The `advfreeze` [3.4.13] command places the file system in a metadata-consistent state and guarantees that it stays that way until thawed. 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.

The file system thaws either by timing out or explicitly with the `advthaw` [3.4.13] command. If you are running a TruCluster configuration, shutting down any node or the failure of any node will also thaw the file system.

## 2.6 Backup Utilities

AdvFS provides enhanced backup utilities that allow remote backup capabilities. These utilities will backup extended file attributes including acls and quotas. These utilities, `advdump` [3.4.16] and `advrdump` allow incremental backups in addition to full backups.

The commands perform either an incremental backup, level 9 to 1, or a full backup, level 0, depending on the desired level of backup and the level of previous backups recorded in the `/etc/advdumpdates` file.

Note that an incremental backup only captures the files that have changed, ignoring all others. This means that if the user performs a level 0 backup and then a later incremental backup, deleted files are not marked as gone (deleted). If the user does a complete restore with a level 0 `saveset` and incremental backups, the deleted files will be restored. These files must be deleted individually.

The commands back up all files that are new or have changed since the latest backup date of all backup levels that are lower than the backup level being performed. If a backup level that is lower than the specified level does not exist, the commands initiate a level 0 backup. A level 0 backup backs up all the files in the file system.

The backup is restored using the `advrestore` [3.4.17] and `advrrestore` commands.

## 2.7 Trashcans

AdvFS has the ability to create trashcans where deleted files are stored for possible recovery. This functionality is available to end-users and system administrators through the `advtrashcan` [3.4.15] command.

## 2.8 On-Disk Structure Viewer

AdvFS provides a viewer, `advvods`, where various AdvFS on-disk structures can be examined.

## 2.9 File System Expansion

AdvFS supports online file system expansion in two ways. Multiple volume support allows additional disk volumes to be added (or removed) at any time. The `extendfs` (and `fsadm`) commands allow the file system to expand on an existing disk volume if it has extra space. Users can manage their storage requirements in whichever manner suits their needs.

## 2.10 Snapshot Support

AdvFS supports the ability to take a snapshot of an existing file system. This snapshot is a moment-in-time read-only copy of the original and can be accessed just like normal read-only AdvFS file systems. Since snapshots are an extension of the original file system, it inherits many properties of the parent, including the

underlying storage. The fsadm command is the interface for managing snapshots. In order to create a snapshot, the AdvFS file system must be named (identified with a /dev/advfs pathname). Once a user creates a snapshot from <fsname>, it can be accessed as /dev/advfs/<fsname>/<snapshot name> for mount and other AdvFS commands.

## 2.11 Storage Thresholds

AdvFs supports the ability to set threshold limits on file systems. A threshold limit can be thought of as a high or low watermark that can be implemented to monitor filesystem storage consumption. The user can set (via fsadm chfs) both an upper and a lower threshold limit for a given file system. When a threshold limit is crossed an EVM event is generated, indicating the type of threshold (upper or lower) crossed and for which filesystem. Upper threshold crossing can be used as an indicator that the file system is consuming storage near it's capacity limit, and lower threshold crossing can be used as an indicator that the file system is under utilizing it's storage capacity. See man pages for 'fsadm chfs' and 'fsadm info'.

## 2.12 Command Summary

<i>Task</i>	<i>Command Syntax</i>
Create file system	mkfs -F advfs [-n <fsname>] <special> newfs -F advfs [-n <fsname>] <special>
Mount file system	mount -F advfs [options] {<special>   <fsname>} <dir>
Unmount file system	umount <dir>
Display general file system info	fsadm -F advfs [options] {<special>   <fsname>}
Add a volume (multi-volume)	fsadm -F advfs addvol <special> <fsname>
Control the background optimization utility	fsadm -F advfs autotune [options] {<special>   <fsname>}
Balance file system (multi-volume)	fsadm -F advfs balance <fsname>
Change properties	fsadm -F advfs chfs [-o <options>] {<special>   <fsname>}
Change I/O properties	fsadm -F advfs chio [options] <special> [<fsname>]
Defragment file system	fsadm -F advfs defrag [options] {<special>   <fsname>}
Extend file system	fsadm -F advfs extend [options] <special> [fsname] extendfs -F advfs [options] <special> [fsname]
Display file attributes and extent maps	fsadm -F advfs getattr [options] <filename>
Display license mode	fsadm -F advfs license
Display file systems and volumes	fsadm -F advfs list [options] [special fsname]
Migrate (multi-volume)	fsadm -F advfs migrate [options] <filename>
Convert single volume to multi volume	fsadm -F advfs multi <special> <fsname>
Switch the log to another volume	fsadm -F advfs mvlog <special> <fsname>
Preallocate file storage	fsadm -F advfs prealloc [-o option_list] file size
Remove file system (including snapshot)	fsadm -F advfs rmfs {<special>   <fsname>}
Remove a volume (multi-volume)	fsadm -F advfs rmvol <special> <fsname>
Locates AdvFS volumes on the system	fsadm -F advfs scan [options] <special>
Display on disk structures	advvods [options] {<special>   <fsname>}
Make, remove, or display trashcan	advtrashcan [options]
Show advfs stats	advstat [options] {<special>   <fsname>}
Backup/restore file system	advdump -F advfs [options] <mount point> advrestore -F advfs [-f <device>] [options]
Recover file data from damaged AdvFS file system	advsalvage [options] {<special>   <fsname>}
Verify quotas	quotacheck -F advfs [options] {<special>   <fsname>}
File system check	fsck -F advfs [options] {<special>   <fsname>}
Pathname check	ncheck -F advfs [options] {<special>   <fsname>}
Summarize file system ownership	quot -F advfs [options] {<special>   <fsname>}
Freeze file system	advfreeze [options] <mount point>

<i>Task</i>	<i>Command Syntax</i>
Thaw file system	advthawfs <mount point>
Edit quotas	edquota -F advfs
File name list with statistics	ff -F advfs [options] {<special>   <fsname>}
Display user block usage info	advdiskusg -F advfs [options] {<special>   <fsname>}
Create a snapshot	fsadm -F advfs snap <fsname> <snapshot name>



## 3 External Reference Specifications

### 3.1 Internal Interfaces

Not Applicable

### 3.2 External Interfaces

Not Applicable

### 3.3 Interface to Programs

#### 3.3.1 Data Structures

The following data structures will be visible to user programs. The file <sys/fs/advfs\_ioctl.h> will contain these data structures along with the interface declarations that are described in the rest of the section.

<sys/fs/advfs\_ioctl.h>:

```
struct extentmapentry {
    uint64_t ext_offset;      /* offset in bytes */
    uint64_t ext_size;       /* size in bytes */
};

struct extentmap {
    uint64_t extmp_arraysize; /* size of array in bytes. IN */
    uint64_t extmp_numextents; /* total number of extents in file. OUT */
    uint64_t extmp_offset;    /* offset of logical extents. IN/OUT */
    struct extentmapentry *extmp_extent; /* extent array. OUT */
};

#define ADVCACHE 0
#define ADVDIRECTIO 1
```

### 3.3.2 advfs ioctl

#### NAME

advfs\_ioctl - AdvFS file system control functions.

#### SYNOPSIS

```
#include <sys/fs/advfs_ioctl.h>

int ioctl(int fildes, int cmd, ... /* arg */);
```

#### OPTIONS

<i>Option</i>	<i>Description</i>
fildes	open file descriptor for an AdvFS file.
cmd	Command specified in <sys/fs/advfs_ioctl.h>
args	args specific to the command

#### DESCRIPTION

The AdvFS ioctl enhancements provide for extended control and information over open files.

The symbolic names for commands and arguments are defined by the <sys/fs/advfs\_ioctl.h> header file.

The enhancements available are:

##### ADV\_GETMAP

The ADV\_GETMAP request gets the sparseness map of the file referred to by the fildes parameter. The argument parameter, taken as a pointer to type struct extentmap, is filled in with data that describes the logical extent map of the file. See [3.3.1] for the structure definition.

This map is useful for determining what parts of a file have allocated storage and what parts have sparse file holes. With knowledge of where the holes exist in a file, a backup program can skip holes and reduce the time to backup the file. If the open file has holes, then multiple extents will be returned in the extent map array. Otherwise, all logically contiguous extents will be collapsed into a single extent.

One extent map entry will be returned for each logical contiguous section of data in the file. Each extent map entry describes allocated storage in the file. The ext\_offset field is the byte offset into the file. The ext\_size field indicates the size of the valid data that starts from that offset. The bytes between the end of one extent map entry and the beginning of the next extent map entry do not have allocated storage. The ioctl will fill in the extentmapentry structures pointed to by extmp\_extent.

The ioctl returns the extentmap structure with several fields filled in. The extmp\_extent array will contain the logical extent map. The extmp\_numextents field will contain the total number of logical extents in the file. The extmp\_offset field will indicate the next logical extent in the file that has not been returned in the map. If the extmp\_offset field is 0, then the last logical extent of the file has been returned in the map.

If the file is currently being written, it is possible for the map returned and the current file extent map to disagree. It is recommended that you use this function only on files that are not being actively written.

#### ADV\_GETCACHEDPOLICY

The ADV\_GETCACHEDPOLICY request gets the cache policy of the file referred to by the fildes parameter. The argument parameter, taken as a pointer to type int, is either filled with ADVDIRECTIO or ADVCACHE. ADVDIRECTIO indicates that the file's cache policy is direct I/O while ADVCACHE indicates that the file's cache policy is the default file system's cache policy.

#### ADVFS\_FREEZEFS

ADVFS\_FREEZEFS causes the specified file system to enter into a metadata stable state. This places the file system metadata in a consistent state and guarantees that it stays that way until thawed. 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.

```
ioctl( (int)fildes, ADVFS_FREEZEFS, (int)timeout );
```

The passed argument is a timeout. If timeout is greater than zero specified the maximum time allowed for the file system to remain frozen. If timeout is zero then use the default timeout as specified by freezefs\_default\_timeout or if freezefs\_default\_timeout is not specified, then default to 60 seconds. If timeout is less than zero, this specifies no timeout and the fileset's file system remains frozen until explicitly thawed by ADVFS\_THAWFS. This applies to both the parent file system and all of its snapshots (if they exist).

#### ADVFS\_FREEZEQUERY

ADVFS\_FREEZEQUERY returns the current freeze state of the specified file system.

```
status = ioctl( (int)fildes, ADVFS_FREEZEQUERY, (int)0 );
```

The return value is zero if the file system is not frozen. The value is greater than zero if the file system is frozen.

#### ADVFS\_THAWFS

ADVFS\_THAWFS causes the previously frozen file system to unfreeze and allow normal I/O activity.

```
ioctl( (int)fildes, ADVFS_THAWFS, (int)0 );
```

## 3.4 Interface to Users

### 3.4.1 newfs\_advfs

#### NAME

`newfs` – Construct a new AdvFS file system

#### SYNOPSIS

```
/usr/sbin/newfs [-F advfs] [-frV] [-B | -R size] [-o option_list] [-n name]
    special
/usr/sbin/newfs [-F advfs] [-mV] special
```

#### OPTIONS

`newfs` recognizes the following options:

<i>Option</i>	<i>Description</i>
<code>-B</code>	Reserve space for boot programs past the end of the file system. If file <code>/usr/lib/uxbootlf</code> is present on the system then sufficient space to accommodate that file is reserved, otherwise 691 KB sectors are reserved. This option decreases the size of the file system to be created.
<code>-F advfs</code>	Specify the AdvFS file system type.
<code>-f</code>	Force creation of the file system on the given device regardless of any checks and warnings.
<code>-o option_list</code> <code>  logsize=blocks</code>  <code>  blksize=blocks</code>  <code>  rootfs</code>	Specify file system specific options in a comma-separated list. Specify the size of the log in <code>DEV_BSIZE</code> blocks. It can also be specified in megabytes with the 'M' suffix. Specify the size of a user data block.  Specify the file system as a boot file system. This only allows a single volume in the filesystem (same as '-r').
<code>-R size</code>	Reserve size megabytes (MB) of swap space past the end of the file system. This option decreases the size of the file system to be created by the given amount.
<code>-r</code>	Specify the file system as a boot file system. This only allows a single volume in the file system.
<code>-n name</code>	Assign an fsname specified by name for the file system.
<code>-m</code>	Print the command used to create the filesystem on <code>special</code> . It will determine the following options: <ul style="list-style-type: none"><li>• <code>-o logsize=blocks</code></li><li>• <code>-o blksize=blocks</code></li><li>• <code>-r</code></li><li>• <code>-n name</code></li></ul>
<code>-V</code>	Echo the complete command line, but perform no action.

#### DESCRIPTION

The `newfs` command is a front-end wrapper for `mkfs`.

`special` represents a block special device.

The `newfs` command creates an AdvFS file system by writing on the `special` device file. If the '-V' option is specified, then `mkfs` will print out the command string used to invoke it and will exit. Otherwise, `mkfs` will read

the super block of the special and check to see if there is valid data. If there is valid data, then mkfs will query the user about overwriting the disk. The '-f' option will override that check. If the '-n' option is specified, then mkfs will use that name when creating the entry in */dev/advfs* using the supplied 'fsname.' 'fsname' must not begin with a period (".").

The user can specify the size of the log with the '-o logsize=' option. A valid size argument will range from 4096 to 524256 blocks. This log size argument will be rounded down to a multiple of 32 blocks. If a size is not specified, then the default log size is calculated with a minimum of 4096 blocks (4 megabytes).

AdvFS multi-volume support requires that the fsname be specified for fsadm operations.

## **RESTRICTIONS**

The user must have root privileges.

## **ERRORS**

## **EXIT STATUS**

0 Success  
>0 An error occurred.

### 3.4.2 mkfs\_advfs

#### NAME

mkfs – Construct a new AdvFS file system

#### SYNOPSIS

```
/usr/sbin/mkfs [-F advfs] [-frV] [-B | -R size] [-o option_list] [-s size]
  [-n name] special
/usr/sbin/mkfs [-F advfs] [-mV] special
```

#### OPTIONS

mkfs recognizes the following options:

<i>Option</i>	<i>Description</i>
-B	Reserve space for boot programs past the end of the file system. If file /usr/lib/uxbootlf is present on the system then sufficient space to accommodate that file is reserved, otherwise 691 KB sectors are reserved. This option decreases the size of the file system to be created.
-F advfs	Specify the AdvFS file system type.
-f	Force creation of the file system on the given device regardless of any checks and warnings.
-o option_list logsize=blocks  blksize=blocks  rootfs	Specify file system specific options in a comma separated list. Specify the size of the log in DEV_BSIZE blocks. It can also be specified in megabytes with the 'M' suffix. Specify the size of a user data block. Specify the file system as a boot file system. This only allows a single volume in the filesystem (same as '-r').
-R size	Reserve size megabytes (MB) of swap space past the end of the file system. This option decreases the size of the file system to be created by the given amount.
-r	Specify the file system as a boot file system. This only allows a single volume in the file system.
-n name	Assign an fsname specified by name for the file system.
-m	Print the command used to create the filesystem on special. It will determine the following options: <ul style="list-style-type: none"><li>• -o logsize=blocks</li><li>• -o blksize=blocks</li><li>• -r</li><li>• -n name</li></ul>
-V	Echo the complete command line, but perform no action.

#### DESCRIPTION

special represents a block special device.

The mkfs command creates an AdvFS file system by writing on the special device file. If the '-V' option is specified, then mkfs will print out the command string used to invoke it and will exit. Otherwise, mkfs will read the super block of the special and check to see if there is valid data. If there is valid data, then mkfs will query the user about overwriting the disk. The '-f' option will override that check. If the '-n' option is specified, then

mkfs will use that name when creating the entry in */dev/advfs* using the supplied 'fsname.' 'fsname' must not begin with a period (".").

The user can specify the size of the log with the '-o logsize=' option. A valid size argument will range from 4096 to 524256 blocks. This log size argument will be rounded down to a multiple of 32 blocks. If a size is not specified, then the default log size is calculated with a minimum of 4096 blocks (4 megabytes).

AdvFS multi-volume support requires that the fsname be specified for fsadm operations.

## **RESTRICTIONS**

The user must have root privileges.

## **ERRORS**

## **EXIT STATUS**

0 Success  
>0 An error occurred.

### 3.4.3 mount\_advfs

#### NAME

mount – mount an AdvFS file system

#### SYNOPSIS

```
/usr/sbin/mount [-F advfs] [-l] [-p | -v]
/usr/sbin/mount [-F advfs] -a [-eQ]
/usr/sbin/mount [-F advfs] [-eQVr] [-o options] {special | fsname}
    directory
/usr/sbin/mount [-F advfs] [-eQVr] [-o options]
    {special | fsname | directory}
```

#### OPTIONS

mount recognizes the following options:

<i>Option</i>	<i>Description</i>
-F advfs	Specify the AdvFS file system type.
-a	Attempt to mount all file systems described in /etc/fstab. All optional fields in /etc/fstab must be included and supported. If -F advfs is specified, all AdvFS file systems in /etc/fstab are mounted. If noauto is specified in an entry's option list, this entry is skipped. File systems are not necessarily mounted in the order listed in /etc/fstab.
-l	Limit actions to local file systems only.
-p	Report the list of mounted file systems in the /etc/fstab format.
-Q	Prevent the display of error messages resulting from an attempt to mount already mounted file systems.
-e	Verbose mode.
-v	Reports the regular output with file system type and flags, however, directory and special fields are reversed.
-V	Echo the completed command line, but perform no other action.
-r	Mount in read-only mode.



<i>Option</i>	<i>Description</i>
-o options rw	mount with read/write access. DEFAULT
ro	mount read only
remount	mount update. Allows changing mount parameters, mntfromname, and status from ro to rw.
suid	Allow set-user-ID execution. DEFAULT
nosuid	Do not allow set-user-ID execution
server_only	CFS argument. Prevents the file system from being read from or written to by other members in the cluster.
atimes	Flushes access times to disks as they occur. DEFAULT
noatimes	Marks file access time changes made for reads of regular files in memory, but does not flush them out to disk until metadata and other file modifications are made.
server=name	CFS argument which indicates the cluster member to initially serve the file system
dirty	Skips recovery when the file system is mounted.
noreadahead	Disables read-ahead functionality on this mount point. By default, read-ahead functionality is enabled.

## DESCRIPTION

mount enables access to an AdvFS file system from a directory on an existing file system. If a snapshot is specified, it is mounted read-only regardless of the rw option.

Note: The '-o dirty' option is unsafe and some metadata changes may be lost.

## RESTRICTIONS

The user must have root privileges.

## ERRORS

## EXIT STATUS

0 Success  
>0 An error occurred.



### 3.4.4 umount\_advfs

#### NAME

umount - unmount mounted file systems

#### SYNOPSIS

```
/usr/sbin/umount [-V] [-v] [-s] { special | directory }  
/usr/sbin/umount [-F advfs] [-v] [-s] [-h host] -a
```

#### OPTIONS

umount recognizes the following options:

<i>Option</i>	<i>Description</i>
-F advfs	Specify the AdvFS file system type.
-a	Attempt to unmount all file systems described in /etc/mnttab. If '-F advfs' is specified, all AdvFS file systems in /etc/mnttab are unmounted.
-v	Verbose mode.
-V	Echo the completed command line, but perform no other action.

#### DESCRIPTION

The umount command unmounts mounted file systems. Use fsname or directory to specify the file system if it is multi-volume.

#### RESTRICTIONS

The user must have root privileges.

#### ERRORS

#### EXIT STATUS

0 Success  
>0 An error occurred.

### 3.4.5 quotacheck\_advfs

#### NAME

quotacheck – file system quota consistency checker

#### SYNOPSIS

```
/usr/sbin/quotacheck [-F advfs] [-V] [-pPv] {special | fsname | mount_dir}  
/usr/sbin/quotacheck [-F advfs] [-V] [-pPv] -a
```

#### OPTIONS

<i>Option</i>	<i>Description</i>
-F advfs	Specify the AdvFS file system type.
-a	Checks file systems listed in the /etc/fstab file.
-V	Echo the completed command line, but perform no action.
-v	Reports discrepancies between the calculated and recorded disk quotas to the user.
-P	noop. Exists for standards compatibility.
-p	noop. Exists for standards compatibility.

#### DESCRIPTION

The quotacheck utility examines the specified file system, building a table of current user disk usage. It then compares this table to the one stored in the user quota file for the file system. If any inconsistencies are detected, both the quota file and the current system copy of the incorrect quotas are updated.

The 'quotacheck -v' command will report inconsistencies to the user terminal along with updating the tables.

#### RESTRICTIONS

The user must have root privileges to run quotacheck.

#### ERRORS

#### EXIT STATUS

0       Success  
>0      An error has occurred.

### 3.4.6 `extendfs_advfs`

#### NAME

`extendfs (advfs)` – extend an AdvFS file system size.

#### SYNOPSIS

```
/usr/sbin/extendfs [-F advfs] [-q | -s size] [-v] special  
/usr/sbin/extendfs [-F advfs] [-q | -s size] [-v] [special] fsname
```

#### OPTIONS

<i>Option</i>	<i>Description</i>
<code>-F advfs</code>	Specify the AdvFS file system type.
<code>-q</code>	Query the size of the underlying volumes. No file system extension will be done.
<code>-v</code>	Verbose flag.
<code>-s size</code>	Specifies the number of <code>DEV_BSIZE</code> blocks to be added to the file system.

#### DESCRIPTION

The `extendfs` command expands the file system on the underlying volume. If the `'-s size'` option is NOT specified, `extendfs` will expand the file system to use all of the underlying storage of the volume. `extendfs` will operate on either an unmounted or mounted AdvFS file system.

The user can extend a single-volume file system, one volume in a multi-volume file system, or all volumes in a multi-volume file system. The user cannot specify a size if all of the volumes in the file system are to be extended. In order to extend all of the volumes to a specific size in a multi-volume file system, `extendfs` has to be executed for each volume. If the file system is multi-volume, then `fsname` needs to be the last argument.

#### RESTRICTIONS

The user must have root privileges to run `extendfs`.

#### ERRORS

#### EXIT STATUS

0       Success  
>0     Error

### 3.4.7 fsadm\_advfs

#### NAME

fsadm – an AdvFS file administration command

#### SYNOPSIS

```
/usr/sbin/fsadm [-F advfs] [info] [-V] [-m] {special | fsname}
/usr/sbin/fsadm [-F advfs] addvol [-V] [-f] special fsname
/usr/sbin/fsadm [-F advfs] autotune [-V] activate|deactivate|suspend|status
    {special | fsname}
/usr/sbin/fsadm [-F advfs] autotune [-V] -l|-L extents|hotfiles
    {special | fsname}
/usr/sbin/fsadm [-F advfs] autotune [-V] -o option_list {special | fsname}
/usr/sbin/fsadm [-F advfs] balance [-V]
/usr/sbin/fsadm [-F advfs] chfs [-V] -o option_list {special | fsname}
/usr/sbin/fsadm [-F advfs] chio [-V] [-l] [-o option_list] special
    [fsname]
/usr/sbin/fsadm [-F advfs] defrag [-V] [-L] [-e filename]
    [-w <timelimit>] {special | fsname}
/usr/sbin/fsadm [-F advfs] extend [-V] [-n size] special
/usr/sbin/fsadm [-F advfs] extend [-V] [-n size] [special] fsname
/usr/sbin/fsadm [-F advfs] getattr [-V] [-I] [-h | -x] filename...
/usr/sbin/fsadm [-F advfs] license
/usr/sbin/fsadm [-F advfs] list [-b|-c|-s] [special|fsname]
/usr/sbin/fsadm [-F advfs] migrate [-V] [-k fileoffset] [-n blkcount]
    [-s special] [-d special] filename
/usr/sbin/fsadm [-F advfs] multi [-V] special fsname
/usr/sbin/fsadm [-F advfs] mvlog [-V] special fsname
/usr/sbin/fsadm [-F advfs] prealloc [-o option_list] filename size
/usr/sbin/fsadm [-F advfs] rmfs [-V] [-f] [-Rr] {special | fsname}
/usr/sbin/fsadm [-F advfs] rmvol [-V] [-f] special fsname
/usr/sbin/fsadm [-F advfs] scan [-V] [-g] [-a] [-o option_list]
    special...
/usr/sbin/fsadm [-F advfs] snap [-V] fsname snapshot_name
```

#### OPTIONS

fsadm uses the following syntax: fsadm keyword options...

These options work with all keywords:

- F Specify the AdvFS file system type.
- V Echo the completed command line and exit.

Valid keywords are:

**info** | **<none>**

Displays information about the file system

Options	
-m	Displays only the metadata size.

**addvol**

Adds special to the file system. Requires multi-volume support.

<i>Options</i>	
-f	Answers yes to any safety prompting.

**autotune**

Controls the background optimization utility. There are control, display, processing and configuration options.

<i>Options</i>	
Control options: activate Activates optimization operation on the file system.  deactivate Deactivates all optimization operations on the file system and stops gathering any internal statistics. This is the default.  status Displays the current optimization configuration, operational statistics, and processing options for the file system.  suspend Temporarily disables all option processing on the file system, but continues to gather internal statistics.	
Display options: -l hotfiles  extents Displays the most actively paging files and the volumes on which they reside. Displays the files currently queued for defragmentation and volume free space balancing.	
-L hotfiles  extents Displays a distribution summary by volume and file system of the most actively paging files. Displays fragmentation summaries by volume.	

**-o option\_list**

Allows the user to change one or more processing options or processing configurations by specifying a comma-separated list.

The following are processing options for the -o option:

**balance=**

Equalizes free space across all volumes in a file system by moving fragmented file between volumes.

enable

Enables free space balancing on the file system.

disable

Disables free space balancing on the file system. This is the default.

**defrag\_all=**

Consolidates all file extents of the file system in the background.

enable

disable

This is the default.

**defrag\_active=**

Consolidates extents of active files in the file system.

enable

Enables file defragmentation on the file system.

disable

Disables file defragmentation on the file system. This is the default.

**topIObalance=**

Distributes files with high I/O among volumes to improve I/O throughput.

enable

disable

This is the default

The following are configuration options for the -o option:

**direct\_io=**

Manages optimization effects on files that are currently enabled for direct I/O.

enable

This is the default.

disable

**steady\_state=**

Specifies the number of hours from the first writable file system mount to the start of top I/O balancing.

**percent\_ios\_when\_busy=**

Limits the share of system I/O that optimization uses to a maximum specified by the percent. However, most optimization processing occurs when devices have no other system I/O. The default is 1 percent. The high limit is 50 percent.

**balance\***

Balances the file system among all of the volumes. Requires multi-volume support.



## chfs

Changes various properties for the file system.

Options	
-o option_list	
where option_list can be:	
blkclear	This ensures that the blocks on disk belonging to the file system will be zero-filled and forced to disk before they are available to files.
noblkclear	This disables blkclear in the file system.
ulimit=percent	Where percent equals percentage of total filesystem storage, that when storage consumption crosses, will generate a notification in the form of an EVM event. A valid ulimit is any integer from 0 to 100. ulimit=0 has the effect of deactivating upper threshold functionality for the given filesystem.
llimit=percent	Where percent equals percentage of total filesystem storage, that when storage deletion crosses, will generate a notification in the form of an EVM event. A valid llimit is any integer from 0 to 100. llimit=0 has the effect of deactivating lower threshold functionality for the given filesystem.
uinterval=minutes	Where minutes equal the number of minutes that must expire before subsequent upper limit threshold EVM events can be generated. A default value of 5 is assigned if no uinterval is given. A valid uinterval is and integer from 0 to 10080. (10080 minutes = 1 week)
linterval=minutes	Where minutes equal the number of minutes that must expire before subsequent lower limit threshold EVM events can be generated. A default value of 5 is assigned if no linterval is given. A valid linterval is and integer from 0 to 10080. (10080 minutes = 1 week)

The properties for a read-only snapshot can not be changed.

## chio\*

Changes various I/O properties for the file system.

Options	
<none>	Displays the default device I/O transfer sizes, in 1 kilobyte (KB) blocks, as given by the disk driver's preferred I/O transfer rate.
-l	Displays the range of I/O transfer sizes, in 1 KB blocks, as calculated by the kernel, based on the disk's geometry:  rblks displays the minimum, the maximum, and the preferred transfer size for reads.  wblks displays the minimum, the maximum, and the preferred transfer size for writes.
-o option_list	
where option_list can be:	
read=blocks	Maximum number of 1 KB blocks to read in an I/O request.

write=blocks

Maximum number of 1 KB blocks to write in an I/O request.

activate

Activates a volume after an incomplete fsadm addvol or fsadm rmvol operation.

This command operates on the device special (volume) in the file system. In a single volume only file system, the only operand is the device special. In a multi-volume enabled file system, the operands are device special and file system name. Changing the I/O properties for a snapshot is the same as changing it for its parent since a parent file system and its snapshots share the same underlying storage.

### defrag

Consolidates file extents on the file system

Options	
-e filename	Defragment specific file.
-L	Display fragmentation summary.
-w <timelimit>	Time limit in minutes for foreground defragment.

Even if a snapshot is specified, its parent file system will also be defragmented. If the '-e filename' option is specified and the filename specifies a file on a snapshot then only the file's extents on the snapshot will be defragmented.

### extend

Expands the file system to use more of the underlying storage of the volume(s)

Options	
-n size	Specify the number of blocks that will be added to the file system.

If the '-n size' option is left out, then the file system will be extended to use all of the available storage. If the '-n size' option is included, then the device special must be included along with the fsname in a multi-volume enabled file system. Specifying a snapshot will extend the parent file system along with the snapshot since they share the underlying storage.

### getattr\*

Displays the attributes of AdvFS directories and files

Options	
-h	Displays the raw extent map for an AdvFS file (including any holes). You cannot use both the -h and the -x options in the same command.
-I	When a filename is a directory, displays attributes for the directory's index file.
-x	Displays the full storage allocation map (extent map) for AdvFS files. You cannot use both the -h and the -x options in the same command.

### license

Displays the current AdvFS licensing mode. Some AdvFS features require premium licensing.

### list

Displays AdvFS file systems by fsname and the volumes used with those file systems. If no options are given, then each AdvFS file systems is listed by fsname.

Options	
-b	Show block special file(s) used by each file system.

-c	Show character special file(s) used by each file system.
-s	Lists any snapshots for each file system.
special	Limit the list to the file system that uses this special file
fsname	Limit the list to the file system with this fsname

### multi

Enables multi-volume support and assigns the specified name to the file system. Once the file system is named, a snapshot can be created.

### migrate\*

Migrate data belonging to filename between volumes in a multi-volume file system. If no options are given, then the whole file is moved to another device

<i>Options</i>	
-k fileoffset	Where to start in the file in 1K offset (auto aligned to 8K boundary).
-n blockcount	How much data to move in 1K blocks (auto aligned to 8K boundary).
-s special	Source volume
-d special	Destination volume

### mvlog\*

Moves the log to special. Requires multi-volume support.

### prealloc\*

Preallocate storage for a file

<i>Options</i>	
-o option_list	where option_list can be: nozero: Do not zero-fill blocks pre-allocated for a file. User must be root. reserveonly: Do not update the file's size along with pre-allocating space

### rmfs

Removes the file system or snapshot specified.

<i>Options</i>	
-f	Answers yes to any safety prompting.
-R	Recursively deletes any snapshots below the specified file system.
-r	Equivalent to -R.

### rmvol\*

Removes special from the file system. Requires multi-volume support.

<i>Options</i>	
-f	Answers yes to any safety prompting.

**scan**

Locates AdvFS volumes on the system.

<i>Options</i>	
-a	Scans all devices found in any <i>/dev/advfs</i> file system as well as those in the command line.
-g	Lists the AdvFS volumes in the order they are found.
-o option_list where option_list is a comma separated list:  name=fsname Fixes the volume count and the links in the <i>/dev/advfs</i> directory for the named file system.  recreate Re-creates missing multi-volume file systems.  undo Will restore AdvFS file systems on devices that were previously removed using <i>fsadm rmfs</i> as long as the metadata associated with those file systems has not been overwritten.	

**snap**

Creates a snapshot of the specified file system. It can only be used on a named AdvFS file system.

\* Option requires the file system to be mounted.

**DESCRIPTION**

*fsadm* is the interface for administrating an AdvFS file system on HP-UX. It will operate on mounted and unmounted systems. However, some of the functionality of *fsadm* requires the file system to be mounted. The AdvFS specific *fsadm* can be invoked either by using '*fsadm -F advfs*' or running *fsadm* on a mounted */etc/fstab* AdvFS file system.

*Displaying file system status*

The user will use the '*fsadm*' command to display the status of the file system. If the '-m' option is specified, then only the metadata size is displayed.

Below is the list of information displayed without the '-m' option.

<i>Field</i>	<i>Description</i>
Name	Name of the file system. If the file system is multi-volume, then the <i>fsname</i> is displayed.
Id	File system unique identifier.
Version	Version number.
Log Blocks	Number of 1K blocks in the transaction log.
Set Name	Name of snapshot, otherwise "default"
Set Tag	Identifier that differentiates between snapshots of the same file system.
Date Created	The date that the file system was created.
Total Blocks	Total number of DEV_BSIZE blocks in file system.
Free Blocks	Total number of DEV_BSIZE blocks available.
Files	Number of files currently present on the file system.
Quota Status	Are quotas enforced?
Blkclear	Is blkclear on or off?
Is Snapshot?	Is this file system a snapshot? (yes or no)
Has Snapshot?	Does this file system have a snapshot? (yes or no)

<i>Field</i>	<i>Description</i>
<b>Volume Info Below (per volume in file system) (only displayed for mounted file systems)</b>	
Volume Name	Full pathname of the volume.
Volume Id	The volume number within the file system. An L next to the number indicates that the volume contains the transaction log.
Size	Size of volume in DEV_BSIZE blocks.
Free	Number of DEV_BSIZE blocks free.
% Used	Percent of the volume used.
Rblks	Maximum number of DEV_BSIZE blocks to read at a time.
Wblks	Maximum number of DEV_BSIZE blocks to write at a time.

There are a few differences if this command is used on a snapshot. The 'Total Blocks' and 'Free Blocks' refer to the storage of the snapshot, its parent file system, and any other snapshots of the parent file system. 'Log Blocks' will be the same value for the parent file system and all of its snapshots.

NOTE: The volume info will only be displayed if the file system is mounted.

### *Changing properties*

fsadm allows the user to change some of the properties of the file system. If the user wants to change general file system properties, then the 'fsadm chfs' command will be used. If the user wants to change I/O properties, then the 'fsadm chio' command will be used. The 'fsadm chio' command will only work on a mounted file system.

### *Multi-volume administration*

AdvFS has the ability to use multiple volumes within a single file system. The file system must be multi-volume enabled to use the multi-volume commands. Refer to the syntax summary for 'fsadm multi'. Once a file system has been made multi-volume available, it can only be referred to using the name specified with the 'fsadm multi' command. Any scripts or entries in /etc/fstab must be updated accordingly.

In order to add another volume to a file system, use the 'fsadm addvol' command. The existing data on the added volume will be deleted. AdvFS does not support a multi-volume boot file system.

'fsadm rrmvol' can be used to remove the volume from a file system. All of the data on that volume to be removed will be migrated to the remaining volumes of that file system. If there is not enough free space on other volumes in the file system to accept the offloaded files from the departing volume, then the utility moves as many files as possible to free space on other volumes. Then a message is sent to the console indicating that there is not enough space to complete the procedure. The files that were not yet moved remain on the original volume.

'fsadm list' with either the -b or -c option can be used to list the volumes that are owned by the AdvFS file systems. If the a user specifies the -s option, then the command will list any snapshots for the AdvFS file systems.

'fsadm balance' evenly distributes the percentage of used space between volumes. This is especially useful when a new volume is just added to the file system.

'fsadm mvlog' moves the transaction log to the specified volume in the file system. Moving the transaction log is typically done to place the log on a faster or mirrored volume.

'fsadm migrate' migrates a file or part of a file to another volume in the file system. This command can be used to balance a file between multiple volumes or move the whole file to another volume. Because there are no

read/write restrictions when using this command, you can migrate a file while users are reading it, writing to it, or both, without disrupting file I/O. File migration is transparent to users.

When you run the 'fsadm migrate' utility with only the '-k' and '-n' options, the utility attempts to allocate destination pages contiguously on one destination volume in the file system. If there are not enough free, contiguous blocks to accomplish the move, the utility then attempts to allocate the pages to the next available blocks on the same volume. If there are not enough free blocks on the same volume, the utility then attempts to move the file to the next available volume or volumes. The utility returns an error diagnostic if it cannot accomplish the move.

### *General Volume Administration*

The fsadm utility can locate AdvFS volumes (raw disks or LVM volumes) that are in AdvFS file systems. This is done with the 'fsadm scan' command.

Given the AdvFS volumes, you can re-create or fix the /dev/advfs directory of a named file system. For example, if you have moved disks to a new system or lost track of where the AdvFS file systems are, you can use this command to locate them.

Another use of the 'fsadm scan' command is to repair AdvFS file systems when you have broken them. For example, if you mistakenly delete the /dev/advfs directory, delete a file system directory in the /dev/advfs directory, or delete links from a file system directory under the /dev/advfs directory, you can use the utility to fix the problem.

The command accepts a list of block special devices and searches to find volumes that are part of an AdvFS file system.

You can run the scan command to automatically rebuild all or part of your /dev/advfs directory or you can rebuild it manually by supplying all the names of the AdvFS volumes in a file system.

The AdvFS volumes are listed as they are grouped in file systems. Set the '-g' option to list the AdvFS volumes in the order they are found.

Run the scan command with the '-o recreate' option to re-create missing file systems from the /dev/advfs directory, missing links, or the entire /dev/advfs directory.

The command displays the date the file system was created, the on-disk structure version, and the last known or current state of the volume.

### *Expanding a file system*

Sometimes the underlying volumes of a file system are expanded after file system creation time. This is typically possible with a volume manager or an advanced hardware storage solution. In order for the file system to take advantage of the extra storage space added, additional metadata must be created. The 'fsadm extend' command will expand the file system over all of the available storage in all of the volumes. If there is a desire to only expand the file system to a value less than the recently increased storage, then the 'fsadm extend -n size' command will extend the file system by the specified size in blocks. Note that the 'fsadm extend -n size' command can only be used on a per volume basis. See the extendfs [3.4.6] command for another interface for expanding the file system.

### *Removing a file system*

The quick way to remove the file system is to use the 'fsadm rmfs' command. If the file system is removed with fsadm and the underlying volumes have not been changed, then it is possible to regain access to the file system again using 'fsadm scan'.

### *Defragmenting a file system*

When a file consists of many non-contiguous file extents, the file is fragmented on the disk. Heavy file fragmentation reduces the read/write performance because more I/O operations are required to access a fragmented file.

The 'fsadm defrag' utility attempts to reduce the number of file extents in a file system by making files contiguous. Defragmenting a file system often makes the free space on a disk contiguous, resulting in less fragmented file allocations in the future.

fsadm has a defragment interface that allows the user to defragment the entire file system or a single file. Defragmenting the entire file system or single file is done in the foreground and can be set with a time limit.

### *Background Optimization*

AdvFS has a utility that optimizes the file system in the background with low overhead. You can access this utility using the 'fsadm autotune' command. It continuously defragments, balances files, consolidates free space and distributes the file I/O over the volumes of a file system. It will process files that are actively being opened and closed. The utility automatically runs during periods of low system demand so that system administrators do not have to take a system offline.

The optimization utility balances the available free space over volumes to maximize file-creation performance. It defragments all active files. The system administrator does not need to perform balance and defragment maintenance tasks. The utility also uses file I/O statistics gathered by the AdvFS kernel to distribute files with high I/O to different volumes in the file system. This functionality will only work with multi-volume AdvFS file systems.

The utility takes into account reserved file access, but it does not move any of the reserved files.

Files or file systems with direct I/O enabled might be moved when consolidating free space.

If optimization is activated and background defragment, balance=, or topIObalance= are enabled on a file system, then the standard AdvFS balance and defragment operations (using fsadm) are not available.

### *Preallocation*

Ordinarily filesystems allocate storage when blocks are first written. As an optimization a user can preallocate storage via the 'fsadm prealloc' command so as to optimize storage allocation, to improve performance when writing to the file, and to guarantee an amount of storage.

The '-o nozero' option further optimizes the allocation by avoiding the initialization of these allocated blocks. This no-zero-fill option is only available to root (super-user).

The '-o reserveonly' option will prevent the file's size from being updated. This effectively *only* reserves storage on disk for a file.

Preallocation is not allowed on a file system that is mounted read-only (including snapshots).

### *Getattr*

'fsadm getattr' displays the attributes of one or more AdvFS files. The command also displays the extent map of each file.

Below are the attributes listed

<i>Attribute</i>	<i>Description</i>
Id	Unique number that identifies the file (in decimal).
MetaVol	The location of the primary metadata of the file expressed as the device name followed by the volume index in parentheses.
PgSz	The page size in 1K blocks.
Pages	The number of pages allocated to the file.
I/O	The type of write requests to the file. <b>user</b> normal  <b>meta</b> write requests are transaction based.
Perf	The efficiency of file-extent allocation, expressed as a percentage of the optimal extent layout. A high percentage indicates that the AdvFS I/O system has achieved optimal efficiency. A low percentage indicates the need for file defragmentation.
File	The name of the directory or file. If the file is a directory with an index and the -i option is not specified, then the term index follows the directory name. If the -i option is specified, then the directory name follows the term index.
<b>xtnt map information (-h or -x)</b>	
fileBlockOff	The starting file offset block of the extent.
fileBlockCnt	The number of file offset blocks in the extent.
Vol	The location of the extent, expressed as a number. Use fsadm for the corresponding volume name.
volBlock	The starting volume block number of the extent.
volBlockCnt	The number of volume blocks in the extent.
extentCnt	The number of extents.

If a directory is specified at the command line it might have an index file associated with it. The attributes for that index file will only be displayed if the '-I' option is specified. Otherwise, the attributes of the directory itself are displayed.

### *Snapshots*

Given a valid file system, 'fsadm snap' will create a snapshot of the specified file system. This snapshot is a moment-in-time copy of the original file system that can be mounted read-only. Two snapshots of the same file system can not have the same name. The 'fsadm rmfs' command will also remove snapshots along with normal file systems. The command has a recursive delete option that will delete the entire snapshot tree below the specified file system. A file system with an existing snapshot can not be removed with 'fsadm rmfs' without using the recursive option (-R).

## RESTRICTIONS

fsadm requires root privileges except for the following command keywords:  
getattr

## ERRORS



## EXIT STATUS

- 0 Success
- >0 An error occurred.

### 3.4.8 fsck\_advfs

#### NAME

fsck - AdvFS file system consistency check and interactive repair

#### SYNOPSIS

```
/usr/sbin/fsck [-F advfs] [-V] [-yY|-nN]
               [-o {(quiet|verbose|debug),msglog=directory}]
               {special | fsname} ...
```

```
/usr/sbin/fsck [-F advfs] [-V] -o full [[-yY|-nN]|-p]
               [-o {activate,nolog,(quiet|verbose|debug),msglog=directory}]
               {special | fsname} ...
```

```
/usr/sbin/fsck [-F advfs] [-V] -o verify [[-yY|-nN]|-p]
               [-o {(quiet|verbose|debug),msglog=directory}]
               {special | fsname} ...
```

```
/usr/sbin/fsck [-F advfs] [-V] -o sync [-yY|-nN]
               [-o {nolog,(quiet|verbose|debug),msglog=directory}]
               {special | fsname} ...
```

```
/usr/sbin/fsck [-F advfs] [-V] -o undo=directory
               [-o {quiet|verbose|debug}]
               {special | fsname} ...
```

#### OPTIONS

<i>Option</i>	<i>Description</i>
-F advfs	Specifies the AdvFS file system type.
-m   -P	The option does not validate the file system, but sanity checks the entered FS/domain/volume name and verifies that the target volume can be read and contains an AdvFS file system. The -P option is identical to -m and is added for compatibility with boot time fsck usage conventions.
-s	noop. Exists for compatibility.
-V	Echo the completed command line (without the -V), but perform no other action.
-n   -N	Specifies that "no" should be answered at all prompts. If used with -m and the activation fails, do a verify check (no fixes).
-y   -Y	Specifies that "yes" should be answered at all prompts. If used with -m and the activation fails, do a full check with fixes.
-p	"Preens" the file system, proceed to process and repair the file system without user interaction. Exits immediately if there is a problem requiring intervention.
-o options	A comma separated list of options. <i>option</i> <i>Description</i>  full Does a detailed check of the file system metadata, reporting and fixing problems. Old metadata is saved before fixes are made and new metadata written (restore old metadata with "undo option").

<i>Option</i>	<i>Description</i>
	<p><b>verify</b> Does a detailed check of the file system metadata. Problems are reported and it displays how it would fix them, however the fixes are not written to the disk.</p> <p><b>quiet</b> fsck will only print error messages.</p> <p><b>verbose</b> Displays progress messages, detailed error messages, fix information and summary messages.</p> <p><b>debug</b> Debug messages are printed out.</p> <p><b>nolog</b> Skip filesystem activation (log playback) prior to checks &amp; fixes.</p> <p><b>activate</b> After repairing what it can, fsck will attempt to activate the file system at the end</p> <p><b>sync</b> Corrects the AdvFS magic number and synchronizes data that is common across volumes.</p> <p><b>msglog=directory</b> Specifies the directory where fsck writes the message log and undo files.</p> <p><b>undo=directory</b> Restores the file system to its previous state using the most recent undo files in the specified directory.</p>

## DESCRIPTION

The fsck utility checks file system's on-disk metadata for consistency. Fsck typically "activates" a file system, that is, replays any transaction log records and processes deferred deletes. Transaction log records and the deferred delete list may exist if a system is brought down without cleanly unmounting its file systems. When run without options or when run with the "-o full" or "-o sync" options, fsck activates the target file system, doing log playback deferred delete processing if necessary. Files system activation may be explicitly skipped via the "-o nolog" option. Note that skipping activation will cause any transaction log records to be deleted without processing which could result in the loss of user data.

Fsck may be run with any number of file system specifiers at the end of the command line. Each specified file system is checked in the order entered using any command line options specified. If fsck is run without a file system specifier, the /etc/fstab file is searched for AdvFS file systems. Each AdvFS file system found is checked serially in the sequence found using any command line options specified.

If run with the "-m" (or -P) option, fsck sanity checks the target file system and exits. This is done by verifying that the entered file system specifier is valid, and that its implied volume(s) can be opened, read, and that they contain an actual AdvFS file system. No further checks of file system metadata are performed and the file system is not activated. Failure of sanity checks results in fsck returning exit code 2, otherwise exit code 0 is returned. This application of "-m" is typical of fsck variants and is normally used during the boot sequence. These same sanity checks run whenever fsck is invoked for FS checking.

The fsck utility can be run with the “-o full” option to do a full check and repair of the file system metadata. When run this way, fsck scans all the on-disk file system metadata structures searching for corruptions or inconsistencies, which it attempts to fix. Where inconsistencies are found they are corrected so that the metadata is made self-consistent. Only file system metadata is examined and repaired; there is currently no way to check or repair the contents of user files. If fsck cannot repair the on-disk structures for a specific file, the file may be truncated, moved, or deleted depending on specific circumstances. The fsck utility will attempt to save as much of a file as possible. Fsck keeps undo files which preserve all the original disk blocks that fsck modifies. Following a run of fsck that changed the metadata, the filesystem may be restored to its pre-fsck state by running fsck again with the '-o undo=' option.

Alternatively, the fsck utility may be run with the “-o verify” option, which performs all file system checks and reports problems just as when run with the “-o full” option, except that it does not modify the file system in any way. Fsck may also be run with the “-o sync” option, in which it synchronizes common metadata between file system volumes, but does not further checks/fixes.

The fsck utility is primarily concerned with fixing problems that have a limited scope. When a large portion of the file system is corrupted, or it finds corruptions it can't fix, there is very little fsck can do. In these cases, it is recommended to restore the data from backup or running the 'advsalvage' utility.

As mentioned above, the fsck utility keeps an undo file, which contains all the original disk blocks that it changes in the metadata. If the file system containing the undo files runs out of space during the fsck run, the user is prompted on how to proceed. Processing can continue without the undo files by adding more space to the file system containing the undo files, or the user can ask the utility to exit. If the results of fsck are undesirable, then the user can undo the changes by running fsck with the '-o undo=' option. This restores the file system to its original state.

If the '-o msglog=' option is not used, the message log and undo files are put in the current working directory. The message log file is named fsck.<fsname>.log and the undo files are named undo.<fsname>.<#> and undoidx.<fsname>.<#>, where # is a number to be appended to a file name to make it unique. The numbers are rotated sequentially from 0 (zero) through 9 if multiple undo files are created for the same file system. The undo file has the same ending number as its corresponding undo index file.

## **RESTRICTIONS**

The user must have root privileges to run fsck.

## **ERRORS**

## **EXIT STATUS**

- |   |   |
|---|---|
| 0 | Success                                 |
| 1 | Unable to repair all found corruptions. |
| 2 | Program or system error                 |

### 3.4.9 ncheck\_advfs

#### NAME

ncheck – generate a list of pathnames for tag numbers

#### SYNOPSIS

```
/usr/sbin/ncheck [-F advfs] [-i numbers] [-V | -a | -s | -m | -r]
    {special | fsname}...
```

```
/usr/sbin/ncheck [-F advfs]
```

#### OPTIONS

<i>Option</i>	<i>Description</i>
-F advfs	Specify the AdvFS file system type.
-i numbers	List only those files with the specified tag numbers
-a	Includes in the list the path names . (dot) and .. (dot dot), which are ordinarily suppressed.
-m	Includes in the mode, UID, and GID of the files. To use this option you must also specify either the -i or the -s option on the command line.
-r	Operate on the raw device (character) of the file system instead of the block special device. Use this option only if the file system is unmounted.
-s	List only the special files and file with set-user-ID mode.
-V	Echo the completed command line, but take no action.

#### DESCRIPTION

ncheck generates a list of pathnames corresponding to tag numbers for files in the specified AdvFS file systems.

The files are listed in order by tag. To sort the list in a more useful format, pipe the output to the sort command.

ncheck will operate either on a mounted or unmounted AdvFS file system.

If ncheck is run without a specified file system, it will run on all AdvFS file systems it finds in /etc/fstab.

#### ERRORS

#### EXIT STATUS

0 Success  
>0 An error occurred.

### 3.4.10 quot\_advfs

#### NAME

quot - summarize file system ownership

#### SYNOPSIS

```
/usr/sbin/quot [-F advfs] [-V] [-cfhnv] {special | fsname | mount_point}
/usr/sbin/quot [-F advfs] [-V] [-cfhnv] -a
```

#### OPTIONS

<i>Option</i>	<i>Description</i>
-F advfs	Specify the AdvFS file system type.
-V	Echo the completed command line, but take no action.
-c	Displays three columns about user files: file size in blocks, number of files that are the size described in column one, and the cumulative total of blocks that are in file of the size or smaller described in column one.
-f	Displays three columns about user files: the space allocated for the user's files in blocks, the number of files owned by the user, and the user name.
-h	Displays two columns about user files: an estimate of the space allocated for the user files in blocks and the user name. The estimate does not include calculations for sparse files and therefore inaccurately represents them.
-n	Accept data from the ncheck command as input. Run the following pipeline: ncheck fsname   sort +0n   quot -n fsname
-v	Displays five columns about user files: the space allocated for the user files in blocks, the user name, and how many blocks there are in user files that have not been accessed in 30, 60, and 90 days.
-a	Generate a report for all AdvFS file systems in /etc/fstab.

#### DESCRIPTION

The quot command displays the number of 1024-byte blocks in the named file system(s) that are currently owned by each user.

The file system must be mounted.

#### ERRORS

#### EXIT STATUS

0 Success  
>0 An error occurred.

### **3.4.11 edquota, repquota, quotaon/quotaoff, quota**

#### **NAME**

There will be no specific AdvFS command. See the HP-UX edquota man page.

### 3.4.12 ff\_advfs

#### NAME

ff - list file names and statistics for a file system

#### SYNOPSIS

```
ff [-F advfs] [-a num] [-c num] [-VilSU] [-i tag-list] [-m num] [-n file]
    [-p prefix] {special | fsname}...
```

#### OPTIONS

<i>Option</i>	<i>Description</i>
-a num	Select a file if the tag has been accessed in num days.
-c num	Select a file if the tag has been changed in num days.
-F advfs	Specify the AdvFS file system type.
-i tag-list	Generate names for any tags specified in the tag-list.
-I	Do not display tag number after pathname
-l	Generate pathnames for file with more than one link.
-m num	Select a file associated with the tag if it has been modified in num days.
-n file	Select a file associated with the tag if it has been modified more recently than the specified file.
-p prefix	Add the specified prefix to each pathname. The default prefix is . (dot).
-s	Write the file size, in bytes, after each pathname.
-u	Write the owner's login name after each pathname.
-V	Echo the completed command line, but perform no other action.

#### DESCRIPTION

ff reads the tag list and directories of each file system and prints tag data for files that match the selection criteria. Output consists of the pathname for each saved tag, plus any other file information requested using the print options below. Output fields are positional. The output is produced in tag order; fields are separated by tabs. The default line produced by ff is:

```
pathname tag-number
```

The maximum information ff displays is:

```
pathname tag-number size owner
```

The num parameter in the options descriptions is a decimal number, where +num means more than num days, -num means less than num days, and num means exactly num days. A day is defined as a 24-hour period.

Note: This is a new utility for AdvFS.

#### ERRORS

#### EXIT STATUS

0 Success  
>0 An error occurred.



### 3.4.13 advfreezefs/advthawfs

#### NAME

advfreezefs, advthawfs – The advfreezefs utility causes the specified AdvFS file system to enter into a metadata stable state. The advthawfs utility causes a previously frozen file system to resume normal activity.

#### SYNOPSIS

```
/usr/sbin/advfreezefs [-t time] mount_point
/usr/sbin/advfreezefs [-q] mount_point

/usr/sbin/advthawfs mount_point
```

#### OPTIONS

<i>Option</i>	<i>Description</i>
-q	Query if the AdvFS file system specified by mount_point is frozen
-t time	Specifies the maximum time allowed for the file system to remain frozen: > 0 specifies the timeout value in seconds. = 0 Uses the default timeout specified in freezefs_default_timeout. If not specified, then advfreezefs uses 60 seconds. < 0 specifies no timeout. The file system remains frozen until it is explicitly thawed by way of advthawfs

#### DESCRIPTION

To allow coherent hardware snapshots in multi-volume file system configurations, file system metadata must be consistent across all volumes when the individual volumes are snapped or cloned. These configurations include both multi-volume AdvFS file systems and multi-volume LVM logical volumes.

The advfreezefs command places the file system in a metadata-consistent state and guarantees that it stays that way until thawed. 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.

The file system thaws either by timing out or explicitly with the advthawfs command. If you are running a TruCluster configuration, shutting down any node or the failure of any node will also thaw the file system.

The /, /usr and /var file systems can not be frozen.

The exact nature of the advfreezefs command depends on whether you are running on a single system or a cluster.

##### Single System

When you freeze a file system, any in-process metadata updates are allowed to finish. File system operations that do not modify metadata are allowed while frozen. Some operations will work normally even if the target file system is frozen, for example:

- + Read
- + Non-extending write
- + Stat

File system operations that modify metadata are blocked. The following operations will block if the target file system is frozen, for example:

- + Remove a volume
- + Extending write

The following file system operations will fail immediately if the target file system is frozen:

- + Mount (remount) of file system
- + Unmount of file system

## Cluster

When you freeze a file system in a clustered configuration, all in-process file system operations are allowed to complete. Some file system operations are allowed while frozen. The following operations, which do not require metadata updates, will work normally even if the target file system is frozen.

- + Read
- + Stat

Most new file system operations are blocked when the file system is frozen. Operations that have the potential for requiring metadata updates will block if the target file system is frozen, for example:

- + Write
- + chmod
- + Link
- + Adding a volume
- + Removing a volume

Some file system operations will fail immediately if the target file system is frozen, for example:

- + Unmount of file system
- + Mount (remount) of file system
- + User-initiated planned relocation
- + User-initiated forced unmount

## RESTRICTIONS

User must have root privileges to run `advfreeze` and `advthaw`.

## ERRORS

## EXIT STATUS

- 0 Success

>0 An error occurred.

### 3.4.14 advstat

#### NAME

advstat - Displays AdvFS performance statistics

#### SYNOPSIS

```
/usr/sbin/advstat [options] [stats-type] {special | filename}
```

#### OPTIONS

<i>Option</i>	<i>Description</i>
-i sec	Specifies the time interval (in seconds) between displays. The advstat command collects and reports information only for the specified interval. If sec is omitted, advstat uses a default interval of one second.
-c count	Specifies the number of reports. If a count of 0 is specified then a count of 1 is assumed. If count is omitted, advstat returns one report.
-s	Displays raw statistics for the interval.
<b>stats-type flags:</b>	
-f 0	Displays all file system vnop statistics for the selected file system.
-f 1	Displays all file system lookup statistics for the selected file system.
-f 2	Displays common file system vnop statistics.
-l 0	Displays basic lock statistics.
-l 1	Displays lock statistics.
-l 2	Displays detailed lock statistics.
-v	Displays volume read/write statistics.
-B r	Displays BMT Record read statistics.
-B w	Displays BMT Record write statistics.

#### DESCRIPTION

The advstat command displays a wide selection of AdvFS performance statistics. It reports in units of DEV\_BSIZE per interval, with the default being one second.

The options '-i' and '-c' require parameters.

Only one stats-type may be specified with the command. The '-f', '-l', '-v', and '-B' stats-type require parameters.

The following tables define the attributes displayed under the given heading:

No options specified:

TBD

#### ERRORS

#### EXIT STATUS

0 Success  
>0 An error occurred.

### 3.4.15 advtrashcan

#### NAME

advtrashcan - Creates, removes, or shows a trashcan directory

#### SYNOPSIS

```
/usr/sbin/advtrashcan -m trashcan directory...  
/usr/sbin/advtrashcan -r directory...  
/usr/sbin/advtrashcan directory...
```

#### OPTIONS

-m make trashcan  
-r remove trashcan

#### DESCRIPTION

The trashcan utilities enable you to create a trashcan directory, which you attach to any number of directories. This trashcan will store the files that are deleted with the unlink system call.

For instance, you can use the advtrashcan utility to attach a trashcan directory called /usr/trashcan to one or more directories; thereafter, when you delete a file from one of the attached directories, the file system moves the file to the /usr/trashcan directory.

If you mistakenly delete a file, use the mv command to return the file from the /usr/trashcan directory to its original directory.

Note that when more than one directory shares an attachment to a trashcan directory, files deleted with the same name will overwrite each other in the trashcan directory.

Note: The directory and trashcan must be on the same file system.

When you enter advtrashcan at the system prompt, the system shows the trashcan directory, if one exists, for the directory you specified.

#### ERRORS

#### EXIT STATUS

0 Success  
>0 An error occurred.

### 3.4.16 advdump

#### NAME

Performs full and incremental backups on AdvFS file systems

#### SYNOPSIS

```
/usr/sbin/advdump -h
/usr/sbin/advdump -V
/usr/sbin/advdump -w
/usr/sbin/advdump [-0..9] [-CDNUquv] [-F num_buffers] [-T tape_num]
    [-b size] [-f device] [-x num_blocks] mount_point
/usr/sbin/advrdump -h
/usr/sbin/advrdump -V
/usr/sbin/advrdump -w
/usr/sbin/advrdump [-0..9] [-CDNUquv] [-F num_buffers] [-T tape_num]
    [-b size] [-f nodename:device] [-x num_blocks] mount_point
```

#### OPTIONS

<i>Option</i>	<i>Description</i>
-b size	Specifies the number of 1 kilobyte (KB) blocks per record in the saveset. The valid range is 2 through 64 blocks; the default is 60 blocks per record. The value of this option also determines the size of the in-memory buffers.
-C	Compresses the data as it is backed up, which minimizes the saveset size.
-D	Performs a level 0 backup on the specified subdirectory. This option overrides any backup level specification in the command. If this option is specified, the AdvFS quota files are not backed up.
-f device	
-f nodename:device	Specifies the destination of the saveset.  For advdump, the local destination can be a device, a file, or, when the dash (-) character is specified, standard output.  For advrdump, the mandatory specification is nodename:device to specify the remote machine name that holds the device or file.
-F num_buffers	Specifies the number of in-memory buffers to use. The valid range is 2 through 64 buffers; the default is 8 buffers. The size of the in-memory buffers is determined by the value of the -b option.
-h	Displays usage help for the command.
-N	Does not rewind the storage device when it is a tape. Use the -N option when you want to dump more than one saveset to a tape.
-q	Displays only error messages; does not display information messages.
-T tape_num	Specifies the starting number for the first tape. The default number is 1. The tape number is used only to prompt the operator to load another tape in the drive.
-u	Updates the /etc/advdumpdates file with a timestamp entry from the beginning of the backup.
-U	Does not unload the storage device when it is a tape.
-v	Displays the names of the files being backed up.
-V	Displays the current version of the command.
-w	Displays the AdvFS file systems that have not been backed up within one week.
-x num_blocks	Specifies an "exclusive or" (XOR) operation each time the blocks specified by

<i>Option</i>	<i>Description</i>
	num_blocks are written to the saveset. The XOR operation is performed on the blocks and the results written to the saveset as an XOR block that immediately follows the blocks. Subsequently, the advrestore command can use this block to recover one of the blocks in the group should a read error occur. The valid range is 1 through 32 blocks; the default is 8 blocks. Using the -x option creates larger savesets and increases the amount of time required to back up a file system, but offers additional protection from saveset errors.
-0..9	Specifies the backup level. The value 0 for this option causes the entire file system to be backed up to the storage device. The default backup level is 9.

## DESCRIPTION

The advdump command backs up files and any associated extended attributes from a single mounted AdvFS file system to a local storage device.

The advrdump command backs up file and any associated extended attributes from a single mounted AdvFS file system to a remote storage device.

The commands back up all files in the specified file system that are new or changed since a certain date and produce a saveset on the storage device. The date is determined by comparing the specified backup level to previous backup levels recorded in the /etc/advdumpdates file. The default storage device for the advdump command is [/dev/tape/tape0\_d1]<sup>1</sup>. You can specify an alternate storage device by using the -f option. There is no default storage device for the advrdump command; it must be specified.

The commands perform either an incremental backup, level 9 to 1, or a full backup, level 0, depending on the desired level of backup and the level of previous backups recorded in the /etc/advdumpdates file.

Note that an incremental dump only captures the files that have changed, ignoring all others. This means that if you perform a level 0 dump and a later incremental dump, deleted files are not marked as gone (deleted). If you then do a complete restore with a level 0 saveset and incremental backups, the deleted files will be restored. You must then delete these files individually.

The commands back up all files that are new or have changed since the latest backup date of all backup levels that are lower than the backup level being performed. If a backup level that is lower than the specified level does not exist, the commands initiate a level 0 backup. A level 0 backup backs up all the files in the file system.

After the backup operation is complete, you can use the 'advrestore -t' command to verify that the backup contains the files you wanted to save. This command lists the name and size of each file in the saveset without restoring them.

When you specify the '-C' option, the commands back up the files with compression. You cannot specify the compression ratio, it is determined by the contents of the dump.

When you specify the '-u' option, the commands enter a time-stamp entry of that file system and its backup level into the /etc/advdumpdates file.

If a AdvFS file system entry with a specific backup level does not already exist in the /etc/advdumpdates file, the commands append the file with a new advdump record; otherwise, the commands overwrite the existing record, changing the backup date to reflect the most current backup session. This occurs after all files in the named file system have been successfully backed up.

---

<sup>1</sup> Subject to change

If you use the '-N' option to advdump more than one saveset to a tape, see the advrestore command for information on restoring a series of savesets from a tape.

If you use either of the commands to back up a file system to an output file that is part of the file system you are backing up, there are two results you should be aware of:

- + That output file could be twice the size it should be.
- + When you restore that output file, you obtain only a partial copy of it.

To inform you of the situation, the commands display a message similar to the following:

```
advdump: /demo/advdump.file is on the same device as /demo, this
advdump: could cause recursive back up problems.
```

```
advdump: Do you want to abort the dump? (yes or no).
```

Typically, you would want to abort the backup operation and select another file on which to back up the file system. However, there may be situations when you do not want to abort the operation. For example, if you are backing up a portion of a file system using the '-D' option, you can store the resulting output file in the same file system in a section not being backed up.

The advdump and advrdump commands back up only mounted AdvFS file systems.

File systems backed up by using the advdump or the advrdump command must be restored by using the advrestore or the advrrestore command. The advdump and advrdump commands are not interchangeable with the dump and rdump commands. Similarly, the advrestore and the advrrestore commands are not interchangeable with the restore and rrestore commands.

The /etc/advdumpdates file is written in ASCII and consists of a single record per line. You must be the root user to update this file or to change any record field.

## **ERRORS**

### **EXIT STATUS**

- 0 Success
- >0 An error occurred.



### 3.4.17 advrestore

#### NAME

advrestore, advrrestore - Restores files from savesets that are produced by advdump and advrdump

#### SYNOPSIS

```
/usr/sbin/advrestore -h
/usr/sbin/advrestore -V
/usr/sbin/advrestore -t [-f device]
/usr/sbin/advrestore -l [-Q] [-f device]
/usr/sbin/advrestore -i [-mqQv] [-f device] [-D path] [-o opt]
/usr/sbin/advrestore -x [-mqQv] [-f device] [-D path] [-o opt] [file...]
/usr/sbin/advrrestore -h
/usr/sbin/advrrestore -V
/usr/sbin/advrrestore -t [-f nodename:device]
/usr/sbin/advrrestore -l [-Q] [-f nodename:device]
/usr/sbin/advrrestore -i [-mqQv] [-f nodename:device] [-D path] [-o opt]
/usr/sbin/advrrestore -x [-mqQv] [-f nodename:device] [-D path] [-o opt]
    [file...]
```

#### OPTIONS

<i>Option</i>	<i>Description</i>
-D path	Specifies the destination path of where to restore the files. Without the -D option, the files are restored to the current directory.
-f device	
-f nodename:device	When an argument follows the -f option, it specifies the name of the storage device that contains the saveset to be restored. The argument replaces the default device [/dev/tape/tape0_d1].  For advrrestore, the mandatory specification is nodename:device to specify the remote machine name that holds the saveset to be restored. There is no default device.
-h	Displays usage help for the command.
-i	Permits interactive restoration of files read from a saveset. After reading directory information from the saveset, the advrestore command provides a shell-like interface that allows you to select the files you want to restore. Some of the following interactive commands require an arg parameter that is a subdirectory or one or more file names. The other interactive commands use the current directory as default when the arg parameter is not specified. Multiple file names can be separated by spaces. Quotes (") can be used around a file name that contains

Option	Description
	<p>space(s). File names containing quotes (") can be specified by preceding the quote with a backslash (\). The interactive commands are explained in the following list:</p> <p><b>add arg</b> Adds the files in the saveset specified by arg to the list of files to be restored. Files on the list of files to be restored are prepended with the * (asterisk) character when they are listed with the ls interactive command.</p> <p><b>cd [arg]</b> Changes the current saveset directory to the directory specified with the arg parameter.</p> <p><b>delete arg</b> Deletes all files and their subdirectories specified by the arg parameter from the list of files to be restored.</p> <p>An expedient way to select wanted files from any directory in a saveset is to add the directory to the list of files to be restored and then delete the ones that are not wanted.</p> <p><b>extract or restore</b> Restores files, previously added by using the add command, to the current destination directory.</p> <p><b>help</b> Displays help information for the interactive commands.</p> <p><b>ls [arg]</b> Lists files in the current saveset directory or the directory specified with the arg parameter. Directory entries are appended with a slash (/) character. Entries that have been marked to be restored are prepended with an asterisk (*) character.</p> <p><b>pwd</b> Writes the path name of the current saveset directory to the standard output device.</p> <p><b>quit or exit</b> Exits immediately, even when the files on the list of files to be restored have not been read.</p> <p><b>sh command</b> Escapes from the shell, runs the system command, then returns to the shell.</p> <p><b>verbose</b> Selects the -v modifier (see the -v option). The name of each file restored from the saveset is written to the standard output device.</p> <p><b>?</b> Displays help information for the interactive commands.</p>

<i>Option</i>	<i>Description</i>
-l	Lists the entire saveset structure.
-m	Does not preserve the owner, group, or modes of each file from the device.
-o <opt>	Specifies the action to take when a file already exists. The options are: <b>yes</b> Overwrites existing files without any query. The default is yes.  <b>no</b> Does not overwrite existing files.  <b>ask</b> Asks whether to overwrite an existing file.
-q	Prints only error messages; does not print information messages.
-Q	Specifies that quota files should not be restored
-t	Lists the names and size (in bytes) of all files contained in a saveset. Exception: the sizes of any AdvFS quota files are not shown.
-v	Writes the name of each file read from the storage device to the standard output device. Without this option the advrestore command does not notify you about progress on reading from the storage device.
-V	Displays the current version for the command.
-x	Extracts a specific file or files from the saveset. Use this command as an alternate to using the add command in interactive mode. The -x option can precede any other options, but the file... list must be the last item on the command line.

## OPERANDS

file...

Specifies the file or files to restore when using the -x option. All other options must precede any file names on the command line.

## DESCRIPTION

The advrestore and advrrestore commands restore data from a saveset previously archived by the advdump command or the advrdump command. The data, which can be restored from a file, a pipe (not applicable for the advrrestore command), or a storage device (typically tape), is written to the specified directory. The default storage device from which files are read is /dev/tape/tape0\_d1. You can use the '-f' option to specify a different device or file. Tape storage devices can contain more than one saveset. The advrestore and advrrestore commands restore any associated extended attributes.

The default directory into which the files are restored is the current directory. You can specify an alternate directory by using the '-D' option.

Use the '-t' option to list the file names and sizes of the files in a saveset without restoring any files.

If the destination file system is AdvFS, and the saveset contains AdvFS file system quotas, the quotas are restored, even when they differ from the quotas of the destination file system. By using the -Q option, -o no, option or -o ask option, you can prevent this behavior.

The advdump and advrdump commands can write many savesets to a tape. If you want to use the advrestore or the advrrestore commands to restore a particular saveset, you must first position the tape to the saveset by using the mt command with the fsf option to move through your tape.

The source directory path from a advdump command line is stored as a string in the header record of the saveset produced. The advrestore command displays this string when it restores the archived saveset. The string truncates at 128 characters. Several advrestore command options, including '-t', '-l', '-i', and '-x', display the source directory path. The command is the exact string from the advdump command: it contains no relative pathname expansions or resolved symbolic links.

Note that an incremental dump only captures the files that have changed, ignoring all others. This means that if you perform a level 0 dump and a later incremental dump, deleted files are not marked as gone (deleted). If you then do a complete restore with a level 0 saveset and incremental backups, the deleted files will be restored. You must then delete these files individually.

Only the root user can restore AdvFS quota files. A warning message is displayed when a non-root user attempts to use the advrestore command to restore AdvFS quota files. Use the -Q option to prevent the restoration of quota files.

## **ERRORS**

### **EXIT STATUS**

- 0 Success
- >0 An error occurred.

### 3.4.18 advdiskusg

#### NAME

advdiskusg - generate AdvFS disk accounting data by user ID

#### SYNOPSIS

```
/usr/sbin/acct/advdiskusg [-p file] [-i fsnlist] {special | fsname}
/usr/sbin/acct/advdiskusg [-s] [-p file] [-i fsnlist] [file]
/usr/sbin/diskusg -F advfs [-p file] [-i fsnlist] {special | fsname}
/usr/sbin/diskusg -F advfs [-s] [-p file] [-i fsnlist] [file]
```

#### OPTIONS

<i>Option</i>	<i>Description</i>
-s	Input data is already in advdiskusg format. advdiskusg combines all records for a single user into one record. If file is specified, then the input is read from it. Otherwise, the input is read from stdin.
-p file	Use file as the password file to generate login names. /etc/passwd is used by default.
-i fsnlist	Ignore the file systems listed in fsnlist, a comma separated list of file systems. They can be either special or fsname.

#### DESCRIPTION

advdiskusg generates intermediate disk accounting information from data in the file, the specified file system, or the standard input if the -s option is specified and file is omitted. advdiskusg outputs lines on the standard :q:output, one line per user, in the following format:

```
uid login #blocks
```

where:

**uid** User's numerical user ID  
**login** User's login name  
**#blocks** Total number of 512 byte disk blocks allocated to this user

The advdiskusg command obtains user login names from the /etc/passwd file, by default, and reports one record per user to standard output. The command usually reads only the tags of the specified file system.

The output of advdiskusg is normally the input to acctdisk, which generates total accounting records that, can be merged with other accounting records. advdiskusg is normally run in dodisk.

#### ERRORS

#### EXIT STATUS

0 Success  
>0 An error occurred.

### 3.4.19 advsalvage

#### NAME

advsalvage - Recover file data from damaged AdvFS file systems.

#### SYNOPSIS

```
/usr/sbin/advsalvage [-V] [-x | -p] [-l] [-S] [-v number]
  [-d time] [-D directory] [-L path] [-o option]
  [-U format[-u [archive]]] {-s special [-s special]... | fsname}
  [snap_id [path]]
```

#### OPTIONS

<i>Options</i>	<i>Description</i>
-d time	Specifies that the utility attempt to recover only those files that have been modified after the time. Specify the time as a decimal number in the following format:  [[CC]YY]MMDDhhmm[.SS]
-D directory	Specifies the path of the directory to which all recovered files are written. If you do not specify a directory, the utility writes recovered files to the current working directory.  The top level of the directory contains a directory named default, which contains the root of the file system. If the file system contain snapshots, there will be a directory for each snapshot named after the snapshot name.
-l	Specifies verbose mode for messages written to the log file. If you specify this option, the utility writes a message to the log file for every file that is encountered during the recovery. The message contains the file name and file status. You specify the path of the log file by using the -L option.  If you do not specify this option, the utility writes a message to the log file only for partially-recovered and unrecovered files.
-L path	Specifies the path of the directory or the file name for the log file you choose to contain messages logged by the utility.  When your path specification includes a log file name, the utility uses that name for the log file.  When your path specification does not include a log file name, the utility places the log file in that directory and names it advsalvage.log.  When you do not specify this option, the utility places the log file in the current working directory and names it advsalvage.log.
-o option	Specifies the action the utility takes when a file being recovered already exists in the directory to which it is to be written. If you do not specify this option, the default action is to overwrite an existing file without querying the user.  yes Overwrite the existing file without querying the user. This is the default action when option is not specified.

no	Do not overwrite the existing file.
ask	Ask the user whether to overwrite the existing file.
-p	<p>Specifies that the utility identifies a partially-recovered file by appending '.partial' to its file name. If you do not use this option, no file name extensions are added to partially-recovered files.</p> <p>Do not use the -p option with the -x option. If you do, an error message is displayed and the utility exits with an exit value of 2.</p>
-s special [-s special]...	<p>Specifies the block device special file names of volumes in the file system, for example, /dev/disk/dsk3c. The utility attempts to recover files only from the volumes you specify.</p> <p>If you do not specify the -s option, you must specify the fsname operand.</p>
-S	<p>Specifies that the utility is to run in sequential search, checking each page on each volume in the file system. This mode of operation will take a long time on large AdvFS file systems.</p> <p>The -S option can be used to recover most files from a file system, which has been damaged from an incorrect execution of the mkfs utility. In some cases, the recovery will need to generate names based on the file's tag number. These cases usually happen in the root directory, because mkfs usually overwrites this directory.</p> <p>When you specify the -S option, there may be a security issue, because the utility could recover old file systems and deleted files.</p>
-U format	Specifies that advsalvage should recover files in an archive format instead of directly to a file system. The only legitimate value for format is <b>tar</b> .
-u [archive]	The advsalvage command uses the next argument as the name of the archive, or /dev/tape/tape0_d1 if archive is not specified. If the name of the file is - (dash), advsalvage writes to standard output.
-V	Displays the current version of the command.
-v number	Specifies the type of messages directed to stdout. If you do not specify this option, the default is to direct only error messages to stdout. If you specify n to be 1, both errors and the names of partially-recovered files are directed to stdout. If you specify n to be 2, error messages and the status of all files as they are recovered are directed to stdout.
-x	Specifies that partially-recoverable files are not to be recovered. If you do not use this option, partially-recoverable file are recovered. Do not use the '-x' option with the '-p' option.

## DESCRIPTION

The advsalvage utility helps you recover file data after an AdvFS file system has become unmountable due to some type of data corruption. Errors that could cause data corruption of a file system include I/O errors in file system metadata, the accidental removal of a volume, or any I/O error that produces a panic.

Use the advsalvage utility as a last resort. You should first repair file system structures by using the fsck utility. If that repair method is unsatisfactory, attempt to recover file system data from backup media. Only if both methods are unsatisfactory should you employ the advsalvage utility.

Note that the advsalvage utility does not modify the source file system in any way.

## **RESTRICTIONS**

User must have root privileges.

## **ERRORS**

## **EXIT STATUS**

- 0 Success
- >0 An error occurred.



### 3.4.20 advvods

#### NAME

advvods – AdvFS on disk structure viewer

#### SYNOPSIS

```
/usr/sbin/advvods {bmt | rbmt} [-v] fsname [snap_id] [-n]
/usr/sbin/advvods {bmt | rbmt} [-v] fsname [snap_id] -L t=tag [-c]
/usr/sbin/advvods {bmt | rbmt} [-v] fsname -l

/usr/sbin/advvods {bmt | rbmt} [-v] volume_id [-a]
/usr/sbin/advvods {bmt | rbmt} [-v] volume_id [-p page] [-n]
/usr/sbin/advvods {bmt | rbmt} [-v] volume_id -p page mcell [-c]
/usr/sbin/advvods {bmt | rbmt} [-v] volume_id -L b=block [-c]
/usr/sbin/advvods {bmt | rbmt} [-v] volume_id -L t=tag [-c]
/usr/sbin/advvods {bmt | rbmt} [-v] volume_id -l
/usr/sbin/advvods {bmt | rbmt} [-v] volume_id -b block [mcell]

/usr/sbin/advvods {bmt | rbmt} [-v] dumpfile [-a]
/usr/sbin/advvods {bmt | rbmt} [-v] dumpfile [-p page] [-n]
/usr/sbin/advvods {bmt | rbmt} [-v] dumpfile -p page mcell [-c]
/usr/sbin/advvods {bmt | rbmt} [-v] dumpfile -L b=block [-c]
/usr/sbin/advvods {bmt | rbmt} [-v] dumpfile -L t=tag [-c]
/usr/sbin/advvods {bmt | rbmt} volume_id dumpfile

/usr/sbin/advvods file fsname [snap_id] filename [-o fob | -a] [-D]
/usr/sbin/advvods file volume_id -b block
/usr/sbin/advvods file dumpfile [-o fob | -a] [-D]
/usr/sbin/advvods file fsname [snap_id] filename dumpfile

/usr/sbin/advvods log fs_id
/usr/sbin/advvods log [-v | -T] fs_id page [record_offset [-c]]
/usr/sbin/advvods log [-v | -T] fs_id { -a | -R }
/usr/sbin/advvods log [-v | -T] fs_id { -s | -e } page_offset
    [record_offset [-c]]
/usr/sbin/advvods log [-v | -T] { fsname | volume_id } -b block
/usr/sbin/advvods log volume_id dumpfile

/usr/sbin/advvods sbm [-C] fs_id
/usr/sbin/advvods sbm { volume_id | dumpfile } -a
/usr/sbin/advvods sbm { volume_id | dumpfile } -p page [entry]
/usr/sbin/advvods sbm { volume_id | dumpfile } -i index
/usr/sbin/advvods sbm { volume_id | dumpfile } -B block
/usr/sbin/advvods sbm volume_id -b block
/usr/sbin/advvods sbm volume_id -d dump_file
```

```

/usr/sbin/advvods {tag | rtag} [-v] { fsname [snap_id] | dumpfile }
[-a]
/usr/sbin/advvods {tag | rtag} [-v] { fsname [snap_id] | dumpfile }
page
/usr/sbin/advvods {tag | rtag} [-v] fsname [snap_id] filename
/usr/sbin/advvods {tag | rtag} [-v] volume_id -b block
/usr/sbin/advvods {tag | rtag} fsname dumpfile

```

## OPTIONS

<i>Option</i>	<i>Description</i>
Bmt	Display pages of an AdvFS BMT file
File	Display pages of an AdvFS file
Log	Display pages of an AdvFS log file
Rbmt	Display pages of an AdvFS RBMT file
Sbm	Display pages of an AdvFS Storage Bitmap (SBM) file
{tag   rtag}	Display pages of an AdvFS tag or root tag file.
-a	Specifies that all the pages be displayed
-b block	Specifies the logical block number of a disk block on an AdvFS file system
-B block	Display the portion of the SBM that maps the specified block.
-c	Follow chains of related metadata. Alternatively, continue to search.
-C	Check the checksum on each page of the SBM.
-D	Specifies that the output be formatted in a directory hierarchy. The default, if this option is not specified, is to format the output as a hexadecimal and ASCII dump
-e	Specifies that the last active record in the log file (the end of the log file) is to be displayed.
-i index	Display the SBM word specified by the index.
-l	Display the deferred delete list of mcells.
-L b=block	Specifies the logical block number of a disk block on an AdvFS file system. When you use this option, the utility searches the specified BMT file for a mcell that has an extent record for a file that contains the specified block.
-L t=tag	Specifies the file tag number. When you use this option, the utility searches the BMT file for a mcell with this tag.
-V	Force the command to interpret the name you supply as a volume name.
-n	Display free mcells.
-o fob	Specified the File Offset Block (fob) to dump when using the file sub-command
-r	By default, the utility opens the file system using block device special files. Specify the '-r' option to operate on the raw device (character device special file) of the file system instead of the block device.
-R	Specifies that the active log records be displayed.
-s	Specifies that the first active record in the log file--the star of the log file--is to be displayed.
-T	Specifies that only a small amount of the log entry will be displayed.
-v	Verbose. Display all the data in the specified metadata.

## OPERANDS

dumpfile

Specifies the name of a file that contains the output from this utility, in the following format:  
[-d] dump\_file

entry

Specifies the index of the SBM word on the page.

filename

Specifies either the file's file system relative path and file name.

file\_tag

Specifies the file's tag number.

fs\_id

Specifies an AdvFS file system using the following format:  
{ fsname | volume\_id | dumpfile }

fsname

Specifies the AdvFS file system using the following format:  
[-r] storage\_domain\_name

mcell

Specifies the number of a metadata cell (mcell) in a BMT or RBMT file.

page

Specifies the file page number of a metadata file.

page\_offset

Specifies the offset relative to the start or the end of the active region in the log file.

record\_offset

Specifies a byte offset in a page of the log file.

snap\_id

Specifies an AdvFS snapshot using the following format:  
-S snap\_name | -t snap\_tag

If not specified, defaults to primary fileset.

special

Specifies the volume name argument as a full path for the volume, for example /dev/dsk/dsk12.

volume\_id

Specifies an AdvFS volume using the following format:  
{ special | filesystem volnum }

volnum

Specifies the volume number within the file system.

## DESCRIPTION

advvods displays AdvFS metadata located at the page specified. This utility is for advanced users.

## RESTRICTIONS

User must have root privileges.

## ERRORS

## EXIT STATUS

0 Success  
>0 An error occurred.

### 3.4.21 advsavemeta

#### NAME

advsavemeta – Make a copy of an AdvFS file system's metadata.

#### SYNOPSIS

```
/usr/sbin/advsavemeta [-LSTtr] {special | fsname} savedir
```

#### OPTIONS

<i>Options</i>	<i>Description</i>
-L	Do not save the file system's log file to the savedir. If this option is not specified, the default is to write the log file to the savedir.
-r	Allows advsavemeta to run on mounted but corrupt file systems by acquiring metadata information from the raw device.
-S	Do not save each volume's storage bitmap (SBM) to the savedir. If this option is not specified, the default is to write each volume's SBM to the savedir.
-T	Do not save the file system's root tag file to the savedir. If this option is not specified, the default is to write the root tag file to the savedir.
-t	Do not save the file system tag files to the savedir. If this option is not specified, the default is to write the file system tag files to the savedir.

#### DESCRIPTION

The advsavemeta utility takes a snapshot of an AdvFS file system's metadata. This command is used for support purposes only. The resulting output can not be used to backup a file system. See 'advdump' for backup and 'advsalvage' for corrupted file systems.

#### RESTRICTIONS

User must have root privileges.

#### ERRORS

#### EXIT STATUS

0 Success  
>0 An error occurred.

## **4 Product Environment**

### **4.1 Assumptions**

Multi-volume support will be implemented. There will be some type of logical volume manager specific code needed by the utilities. The specific code exists either in the commands itself or a helper library (libfilesys on Tru64). These commands will need to be TruCluster friendly.

The assumption so far is that AdvFS will keep the current infrastructure with domains and filesets. The user will never see the concept of domains and filesets through the user interface. Therefore, many of the AdvFS library calls used by the CLI can be ported to HP-UX without many changes.

HP-UX uses a generic wrapper that is the entry point to many file system specific commands. These include mount, umount, quotacheck, mkfs, newfs, etc. The plan is for AdvFS commands to fit into that model.

## **5 Issues**

### **5.1 Open Issues**

#### **5.1.1 fsadm (migrate) options**

Should the user have the option to specify page offsets and page counts or block offsets and block counts?

STATUS: open

#### **5.1.2 Property list API**

There is a need to expose a property list API. One consumer is CIFS NT acs.

STATUS: open

#### **5.1.3 Management API**

Will management tools need an API to accomplish the same tasks as the CLI?

STATUS: open

## Appendix A: Migration from Tru64 to HP-UX

Below is listed a table that maps Tru64 UNIX AdvFS commands to HP-UX:

<i>Tru64</i>		<i>HP-UX</i>
addvol	→	fsadm
advfsd	→	N/A
advfsstat	→	advstat
advscan	→	fsadm
balance	→	fsadm
chfile	→	N/A
chfsets	→	fsadm
chvol	→	fsadm
clonefset	→	fsadm
defragment	→	fsadm
diskusg	→	advdiskusg
dmapi		N/A
dtadvfs	→	N/A
edquota	→	edquota
fixfdmn	→	fsck
freezefs/thawfs	→	advfreezefs/advthawfs
lmsa	→	N/A
migrate	→	fsadm
mkfdmn	→	mkfs
mkfset	→	mkfs
mktrashcan	→	advtrashcan
mountlist	→	mount
mount	→	mount
mount -o extend	→	extendfs OR fsadm
N/A	→	ff
ncheck	→	ncheck
nvbmtpg	→	advvods
nvfragpg	→	N/A
nvlogpg	→	advvods
nvtagpg	→	advvods
quot	→	quot
quota	→	quota
quotacheck	→	quotacheck
quotaoff	→	quotaoff
quotaon	→	quotaon
renamefset	→	N/A
repquota	→	repquota
rmfdmn	→	fsadm
rmfset	→	fsadm
rmtrashcan	→	advtrashcan
rmvol	→	fsadm
salvage	→	advsalvage
savemeta	→	advsavemeta
shblk	→	N/A
shfragbf	→	N/A
showfdmn	→	fsadm



<i>Tru64</i>		<i>HP-UX</i>
showfile	→	fsadm
showfsets	→	N/A
shtrashcan	→	advtrashcan
stripe	→	N/A
switchlog	→	fsadm
tag2name	→	ncheck
umount	→	umount
vdf	→	N/A
vdump	→	advdump
verify	→	fsck
vfast	→	fsadm
vrestore	→	advrestore
vsbmg	→	advvods