NAME

ufsrestore – incremental file system restore

SYNOPSIS

/usr/lib/fs/ufs/ufsrestore *options* [*arguments*] [*filename* . . .]

DESCRIPTION

ufsrestore restores files from backup media created with the **ufsdump** command. *options* is a single string of one-letter options. One (and only one) of the following options must be included: **i**, **r**, **R**, **t**, or **x**. *arguments* is one or more strings following *options*. The association of arguments with options is determined by order. That is, the first argument goes with the first option that takes an argument; the second argument goes with the second option that takes an argument, and so on. However, the *filename* arguments, which go with either the **x** or **t** options, must come last. They specify the names of files (or directories whose files) are to be restored to disk. Unless the **h** modifier is also used, a directory name refers to the files it contains, and (recursively) its subdirectories and the files they contain.

OPTIONS

Choose One

One of the following options is required:

i Interactive. After reading in the directory information from the media, **ufsrestore** invokes an interactive interface that allows you to browse through the dump file's directory hierarchy and select individual files to be extracted. See **Interactive Commands**, below, for a description of available commands.

r Recursive. Restore the entire contents of the media into the current directory (which should be the top-level of the file system). To completely restore a file system, use this option to restore the level **0** dump, and again for each incremental dump. Although, this option is intended for a complete restore onto a clear file system, if the file system contains files not on the media, they are preserved.

R Resume restoring. **ufsrestore** requests a particular volume of a multi-volume set from which to resume a full restore (see the **r** option above). This allows **ufsrestore** to start from a checkpoint when it is interrupted in the middle of a full restore.

t Table of contents. List each *filename* that appears on the media. If no *filename* argument is given, the root directory is listed. This results in a list of all files on the media, unless the **h** modifier is in effect. The table of contents is taken from the media or from the specified archive file, when the **a** option is used. This option is mutually exclusive with the **x** and **r** options.

x Extract the named files from the media. If a named file matches a directory whose contents were written onto the media, and the **h** modifier is not in effect, the directory is recursively extracted. The owner, modification time, and mode are restored (if possible). Existing files are overwritten and a warning is given. If no *filename* argument is given, the root directory is extracted. This results in the entire tape being extracted unless the **h** modifier is in effect.

Choose Any

In addition to one of the above options, any of the following options may be used:

a *archive_file*
: Read the table of contents from *archive_file* instead of the media. This option can be used in combination with the **t**, **i**, or **x** options, making it possible to check whether files are on the media without having to mount the media. When used with the **x** and interactive (**i**) options, it prompts for the volume containing the file(s) before extracting them.

b *factor*
: Blocking factor. Specify the blocking factor for tape reads. For variable length SCSI tape devices, unless the data was written with the default blocking factor, a blocking factor at least as great as that used to write the tape must be used; otherwise, an error will be generated. Note that a tape block is 512 bytes. Refer to the man page for your specific tape driver for the maximum blocking factor.

c
: Convert the contents of the media in 4.1BSD format to the new **ufs** file system format.

d
: Debug. Turn on debugging output.

f *dump_file*
: Use *dump_file* instead of **/dev/rmt/0** as the file to restore from. Typically *dump_file* specifies a tape or diskette drive. If *dump_file* is specified as '−', **ufsrestore** reads from the standard input. This allows, **ufsdump**(1M) and **ufsrestore** to be used in a pipeline to copy a file system:

example# ufsdump 0f − /dev/rdsk/c0t0d0s7 | (cd /home;ufsrestore xf −)

If the name of the file is of the form *machine***:***device*, the restore is done from the specified machine over the network using **rmt**(1M). Since **ufsrestore** is normally run by root, the name of the local machine must appear in the **/.rhosts** file of the remote machine. If the file is specified as *user@machine***:***device*, **ufsrestore** will attempt to execute as the specified user on the remote machine. The specified user must have a **.rhosts** file on the remote machine that allows the user invoking the command from the local machine to access the remote machine.

h
: Extract or list the actual directory, rather than the files that it references. This prevents hierarchical restoration of complete subtrees from the tape.

m
: Extract by inode numbers rather than by filename to avoid regenerating complete pathnames. Regardless of where the files are located in the dump hierarchy, they are restored into the current directory and renamed with their inode number. This is useful if only a few files are being extracted.

s *n*
: Skip to the *n*'th file when there are multiple dump files on the same tape. For example, the command:

example# ufsrestore xfs /dev/rmt/0hn 5

would position you at the fifth file on the tape.

v
: Verbose. **ufsrestore** displays the name and inode number of each file it restores, preceded by its file type.

y Do not ask whether to abort the restore in the event of tape errors. **ufsrestore** tries to skip over the bad tape block(s) and continue as best it can.

Interactive Commands

ufsrestore enters interactive mode when invoked with the **i** option. Interactive commands are reminiscent of the shell. For those commands that accept an argument, the default is the current directory. The interactive options are:

add [*filename*]
Add the named file or directory to the list of files to extract. If a directory is specified, add that directory and its files (recursively) to the extraction list (unless the **h** modifier is in effect).

cd *directory*
Change to *directory* (within the dump file).

delete [*filename*]
Delete the current directory, or the named file or directory from the list of files to extract. If a directory is specified, delete that directory and all its descendents from the extraction list (unless the **h** modifier is in effect). The most expedient way to extract a majority of files from a directory is to add that directory to the extraction list, and then delete specific files to omit.

extract Extract all files on the extraction list from the dump media. **ufsrestore** asks which volume the user wishes to mount. The fastest way to extract a small number of files is to start with the last volume and work toward the first.

help Display a summary of the available commands.

ls [*directory*]
List files in *directory* or the current directory, represented by a '**.**' (period). Directories are appended with a '/' (slash). Entries marked for extraction are prefixed with a '*' (asterisk). If the verbose option is in effect, inode numbers are also listed.

pwd Print the full pathname of the current working directory.

quit **ufsrestore** exits immediately, even if the extraction list is not empty.

setmodes
Prompts: **set owner/mode for** '**.**' (period). Type **y** for yes to set the mode (permissions, owner, times) of the current directory '**.**' (period) into which files are being restored equal to the mode of the root directory of the file system from which they were dumped. Normally, this is what you want when restoring a whole file system, or restoring individual files into the same locations from which they were dumped. Type **n** for no, to leave the mode of the current directory unchanged. Normally, this is what you want when restoring part of a dump to a directory other than the one from which the files were dumped.

verbose
Toggle the status of the **v** modifier. While **v** is in effect, the **ls** command lists the inode numbers of all entries, and **ufsrestore** displays information about each file as it is extracted.

what Display the dump header on the media.

FILES

/dev/rmt/0	the default tape drive
/tmp/rstdir*	file containing directories on the tape
/tmp/rstmode*	owner, mode, and timestamps for directories
./restoresymtable	information passed between incremental restores

SEE ALSO

mkfs(1M), **mount**(1M), **rmt**(1M), **ufsdump**(1M)

DIAGNOSTICS

ufsrestore complains about bad option characters.

Read errors result in complaints. If **y** has been specified, or the user responds **y**, **ufsrestore** will attempt to continue.

If the dump extends over more than one tape, **ufsrestore** asks the user to change tapes. If the **x** or **i** option has been specified, **ufsrestore** also asks which volume the user wishes to mount.

There are numerous consistency checks that can be listed by **ufsrestore**. Most checks are self-explanatory or can "never happen". Common errors are given below.

Converting to new file system format
A dump tape created from the old file system has been loaded. It is automatically converted to the new file system format.

filename: **not found on tape**
The specified file name was listed in the tape directory, but was not found on the tape. This is caused by tape read errors while looking for the file, or from using a dump tape created on an active file system.

expected next file *inumber*, **got** *inumber*
A file that was not listed in the directory showed up. This can occur when using a dump tape created on an active file system.

Incremental tape too low
When doing an incremental restore, a tape that was written before the previous incremental tape, or that has too low an incremental level has been loaded.

Incremental tape too high
When doing incremental restore, a tape that does not begin its coverage where the previous incremental tape left off, or one that has too high an incremental level has been loaded.

media read error: invalid argument
Blocking factor specified for read is smaller than the blocking factor used to write data.

Tape read error while restoring *filename*
Tape read error while skipping over inode inumber
Tape read error while trying to resynchronize
A tape read error has occurred
If a file name is specified, then its contents are probably partially wrong. If an inode is being skipped or the tape is trying to resynchronize, then no extracted

files have been corrupted, though files may not be found on the tape.

resync ufsrestore, skipped *num*

After a tape read error, **ufsrestore** may have to resynchronize itself. This message lists the number of blocks that were skipped over.

NOTES

ufsrestore can get confused when doing incremental restores from dump tapes that were made on active file systems.

A “level **0**” dump must be done after a full restore. Because **ufsrestore** runs in user mode, it has no control over inode allocation. This means that **ufsrestore** repositions the files, although it does not change their contents. Thus, a full dump must be done to get a new set of directories reflecting the new file positions, so that later incremental dumps will be correct.