

NAME

dump, **rdump** - file system backup

SYNOPSIS

dump [-0123456789acLnrRSu] [-**B** *records*] [-**b** *blocksize*] [-**C** *cachesize*] [-**D** *dumpdates*] [-**d** *density*]
[-**f** *file* | -**P** *pipecommand*] [-**h** *level*] [-**s** *feet*] [-**T** *date*].*filesystem*
dump -**W** | -**w**

DESCRIPTION

The **dump** utility examines files on a file system and determines which files need to be backed up. These files are copied to the given disk, tape or other storage medium for safe keeping (see the **-f** option below for doing remote backups). A dump that is larger than the output medium is broken into multiple volumes. On most media the size is determined by writing until an end-of-media indication is returned. This can be enforced by using the **-a** option.

On media that cannot reliably return an end-of-media indication (such as some cartridge tape drives) each volume is of a fixed size; the actual size is determined by the tape size and density and/or **-B** options. By default, the same output file name is used for each volume after prompting the operator to change media.

The file system to be dumped is specified by the argument *filesystem* as either its device-special file or its mount point (if that is in a standard entry in */etc/fstab*).

dump may also be invoked as **rdump**. The 4.3BSD option syntax is implemented for backward compatibility, but is not documented here.

The following options are supported by **dump**:

-0-9 Dump levels. A level 0, full backup, guarantees the entire file system is copied (but see also the **-h** option below). A level number above 0, incremental backup, tells dump to copy all files new or modified since the last dump of any lower level. The default level is 0.

-a "auto-size". Bypass all tape length considerations, and enforce writing until an end-of-media indication is returned. This fits best for most modern tape drives. Use of this option is particularly recommended when appending to an existing tape, or using a tape drive with hardware compression (where you can never be sure about the compression ratio).

-B *records*

The number of kilobytes per output volume, except that if it is not an integer multiple of the output block size, the command uses the next smaller such multiple. This option overrides the

calculation of tape size based on length and density.

-b *blocksize*

The number of kilobytes per output block. The default block size is 10.

-C *cache size*

Specify the cache size in megabytes. This will greatly improve performance at the cost of **dump** possibly not noticing changes in the file system between passes unless a snapshot is being used. The potential for performance improvement indicates that use of this option together with snapshots is the recommended course of action. Beware that **dump** forks, and the actual memory use may be larger than the specified cache size. The recommended cache size is between 8 and 32 (megabytes).

-c Change the defaults for use with a cartridge tape drive, with a density of 8000 bpi, and a length of 1700 feet.

-D *dumpdates*

Specify an alternate path to the *dumpdates* file. The default is */etc/dumpdates*.

-d *density*

Set tape density to *density*. The default is 1600BPI.

-f *file* Write the backup to *file*; *file* may be a special device file like */dev/sa0* (a tape drive), */dev/fd1* (a floppy disk drive), an ordinary file, or '-' (the standard output). Multiple file names may be given as a single argument separated by commas. Each file will be used for one dump volume in the order listed; if the dump requires more volumes than the number of names given, the last file name will be used for all remaining volumes after prompting for media changes. If the name of the file is of the form "host:file", or "user@host:file", **dump** writes to the named file on the remote host using *rmt(8)*. The default path name of the remote *rmt(8)* program is */etc/rmt*; this can be overridden by the environment variable *RMT*.

-P *pipecommand*

Use *popen(3)* to execute the *sh(1)* script string defined by *pipecommand* for the output device of each volume. This child pipeline's *stdin (/dev/fd/0)* is redirected from the **dump** output stream, and the environment variable *DUMP_VOLUME* is set to the current volume number being written. After every volume, the writer side of the pipe is closed and *pipecommand* is executed again. Subject to the media size specified by **-B**, each volume is written in this manner as if the output were a tape drive.

-h *level*

Honor the user "nodump" flag (UF_NODUMP) only for dumps at or above the given *level*. The default honor level is 1, so that incremental backups omit such files but full backups retain them.

- L** This option is to notify **dump** that it is dumping a live file system. To obtain a consistent dump image, **dump** takes a snapshot of the file system in the *.snap* directory in the root of the file system being dumped and then does a dump of the snapshot. The snapshot is unlinked as soon as the dump starts, and is thus removed when the dump is complete. This option is ignored for unmounted or read-only file systems. If the *.snap* directory does not exist in the root of the file system being dumped, a warning will be issued and the **dump** will revert to the standard behavior. This problem can be corrected by creating a *.snap* directory in the root of the file system to be dumped; its owner should be "root", its group should be "operator", and its mode should be "0770".
- n** Whenever **dump** requires operator attention, notify all operators in the group "operator" by means similar to a wall(1).
- r** Be rsync-friendly. Normally dump stores the date of the current and prior dump in numerous places throughout the dump. These scattered changes significantly slow down rsync or another incremental file transfer program when they are used to update a remote copy of a level 0 dump, since the date changes for each dump. This option sets both dates to the epoch, permitting rsync to be much more efficient when transferring a dump file. The **-r** option can be used only to create level 0 dumps. A dump using the **-r** option cannot be used as the basis for a later incremental dump.
- R** Be even more rsync-friendly. This option disables the storage of the actual inode access time (storing it instead as the inode's modified time). This option permits rsync to be even more efficient when transferring dumps generated from filesystems with numerous files which are not changing other than their access times. The **-R** option also sets **-r**. The **-R** option can be used only to create level 0 dumps. A dump using the **-R** option cannot be used as the basis for a later incremental dump.
- S** Display an estimate of the backup size and the number of tapes required, and exit without actually performing the dump.
- s *feet*** Attempt to calculate the amount of tape needed at a particular density. If this amount is exceeded, **dump** prompts for a new tape. It is recommended to be a bit conservative on this option. The default tape length is 2300 feet.
- T *date*** Use the specified date as the starting time for the dump instead of the time determined from

looking in the *dumpdates* file. The format of date is the same as that of *ctime(3)*. This option is useful for automated dump scripts that wish to dump over a specific period of time. The **-T** option is mutually exclusive from the **-u** option.

- u** Update the *dumpdates* file after a successful dump. The format of the *dumpdates* file is readable by people, consisting of one free format record per line: file system name, increment level and *ctime(3)* format dump date. There may be only one entry per file system at each level. The *dumpdates* file may be edited to change any of the fields, if necessary. The default path for the *dumpdates* file is */etc/dumpdates*, but the **-D** option may be used to change it.
- W** Tell the operator what file systems need to be dumped. This information is gleaned from the files *dumpdates* and */etc/fstab*. The **-W** option causes **dump** to print out, for each file system in the *dumpdates* file the most recent dump date and level, and highlights those file systems that should be dumped. If the **-W** option is set, all other options are ignored, and **dump** exits immediately.
- w** Is like **-W**, but prints only those file systems which need to be dumped.

Directories and regular files which have their "nodump" flag (UF_NODUMP) set will be omitted along with everything under such directories, subject to the **-h** option.

The **dump** utility requires operator intervention on these conditions: end of tape, end of dump, tape write error, tape open error or disk read error (if there are more than a threshold of 32). In addition to alerting all operators implied by the **-n** key, **dump** interacts with the operator on *dump*'s control terminal at times when **dump** can no longer proceed, or if something is grossly wrong. All questions **dump** poses *must* be answered by typing "yes" or "no", appropriately.

Since making a dump involves a lot of time and effort for full dumps, **dump** checkpoints itself at the start of each tape volume. If writing that volume fails for some reason, **dump** will, with operator permission, restart itself from the checkpoint after the old tape has been rewound and removed, and a new tape has been mounted.

The **dump** utility tells the operator what is going on at periodic intervals (every 5 minutes, or promptly after receiving SIGINFO), including usually low estimates of the number of blocks to write, the number of tapes it will take, the time to completion, and the time to the tape change. The output is verbose, so that others know that the terminal controlling **dump** is busy, and will be for some time.

In the event of a catastrophic disk event, the time required to restore all the necessary backup tapes or files to disk can be kept to a minimum by staggering the incremental dumps. An efficient method of staggering incremental dumps to minimize the number of tapes follows:

- Always start with a level 0 backup, for example:

```
/sbin/dump -0u -f /dev/nsa0 /usr/src
```

This should be done at set intervals, say once a month or once every two months, and on a set of fresh tapes that is saved forever.

- After a level 0, dumps of active file systems (file systems with files that change, depending on your partition layout some file systems may contain only data that does not change) are taken on a daily basis, using a modified Tower of Hanoi algorithm, with this sequence of dump levels:

```
3 2 5 4 7 6 9 8 9 9 ...
```

For the daily dumps, it should be possible to use a fixed number of tapes for each day, used on a weekly basis. Each week, a level 1 dump is taken, and the daily Hanoi sequence repeats beginning with 3. For weekly dumps, another fixed set of tapes per dumped file system is used, also on a cyclical basis.

After several months or so, the daily and weekly tapes should get rotated out of the dump cycle and fresh tapes brought in.

ENVIRONMENT

TAPE The *file* or device to dump to if the **-f** option is not used.

RMT Pathname of the remote `rmt(8)` program.

RSH Pathname of a remote shell program, if not `rsh(1)`.

FILES

/dev/sa0 default tape unit to dump to
/etc/dumpdates dump date records (this can be changed; see the **-D** option)
/etc/fstab dump table: file systems and frequency
/etc/group to find group *operator*

EXIT STATUS

Dump exits with zero status on success. Startup errors are indicated with an exit code of 1; abnormal termination is indicated with an exit code of 3.

EXAMPLES

Dumps the `/u` file system to DVDs using **growisofs**. Uses a 16MB cache, creates a snapshot of the dump, and records the *dumpdates* file.

```
/sbin/dump -0u -L -C16 -B4589840 -P `growisofs -Z /dev/cd0=/dev/fd/0` /u
```

DIAGNOSTICS

Many, and verbose.

SEE ALSO

chflags(1), fstab(5), restore(8), rmt(8)

HISTORY

A **dump** utility appeared in Version 4 AT&T UNIX.

BUGS

Fewer than 32 read errors on the file system are ignored, though all errors will generate a warning message. This is a bit of a compromise. In practice, it is possible to generate read errors when doing dumps on mounted partitions if the file system is being modified while the **dump** is running. Since dumps are often done in an unattended fashion using cron(8) jobs asking for Operator intervention would result in the **dump** dying. However, there is nothing wrong with a dump tape written when this sort of read error occurs, and there is no reason to terminate the **dump**.

Each reel requires a new process, so parent processes for reels already written just hang around until the entire tape is written.

The **dump** utility with the **-W** or **-w** options does not report file systems that have never been recorded in the *dumpdates* file, even if listed in */etc/fstab*.

It would be nice if **dump** knew about the dump sequence, kept track of the tapes scribbled on, told the operator which tape to mount when, and provided more assistance for the operator running restore(8).

The **dump** utility cannot do remote backups without being run as root, due to its security history. This will be fixed in a later version of FreeBSD. Presently, it works if you set it setuid (like it used to be), but this might constitute a security risk.