



Full credit is given to the above companies including the OS that this PDF file was generated!

Rocky Enterprise Linux 9.2 Manual Pages on command 'xfsdump.8'

\$ man xfsdump.8

xfsdump(8) System Manager's Manual xfsdump(8)

NAME

xfsdump - XFS filesystem incremental dump utility

SYNOPSIS

xfsdump -h

xfsdump [options] -f dest [-f dest ...] filesystem

xfsdump [options] - filesystem

xfsdump -l [subopt=value ...]

DESCRIPTION

xfsdump backs up files and their attributes in a filesystem. The files are dumped to storage media, a regular file, or standard output. Options allow the operator to have all files dumped, just files that have changed since a previous dump, or just files contained in a list of pathnames.

The xfsrestore(8) utility re-populates a filesystem with the contents of the dump.

Each invocation of xfsdump dumps just one filesystem. That invocation is termed a dump session. The dump session splits the filesystem into

one or more dump streams, one per destination. The split is done in filesystem inode number (ino) order, at boundaries selected to equalize the size of each stream. Furthermore, the breakpoints between streams may be in the middle of very large files (at extent boundaries) if necessary to achieve reasonable stream size equalization. Each dump stream can span several media objects, and a single media object can contain several dump streams. The typical media object is a tape cartridge. The media object records the dump stream as one or more media files. A media file is a self-contained partial dump, intended to minimize the impact of media dropouts on the entire dump stream at the expense of increasing the time required to complete the dump. By default only one media file is written unless a media file size is specified using the `-d` option. Other techniques, such as making a second copy of the dump image, provide more protection against media failures than multiple media files will.

`xfsdump` maintains an online dump inventory in `/var/lib/xfsdump/inventory`. The `-l` option displays the inventory contents hierarchically. The levels of the hierarchy are: filesystem, dump session, stream, and media file.

The options to `xfsdump` are:

`-a` Specifies that files for which the Data Migration Facility (DMF) has complete offline copies (dual-state files) be treated as if they were offline (OFL). This means that the file data will not be dumped by `xfsdump`, resulting in a smaller dump file. If the file is later restored the file data is still accessible through DMF. If both '`-a` option' and '`-z` option' are specified, the '`-a` option' takes precedence (see '`-z` option' below).

`-b` blocksize

Specifies the blocksize, in bytes, to be used for the dump. The same blocksize must be specified to restore the tape. If the `-m` option is not used, then `-b` does not need to be specified. Instead, a default blocksize of 1Mb will be used.

`-c` progname

Use the specified program to alert the operator when a media change is required. The alert program is typically a script to send a mail or flash a window to draw the operator's attention.

-d filesize

Specifies the size, in megabytes, of dump media files. If not specified, xfsdump will dump data to tape using a single media file per media object. The specified media file size may need to be adjusted if, for example, xfsdump cannot fit a media file onto a single tape.

-e Allow files to be excluded from the dump. This will cause xfsdump to skip files which have the "no dump" file attribute set. See the "Excluding individual files" section below for details on setting this file attribute.

-f dest [-f dest ...]

Specifies a dump destination. A dump destination can be the path? name of a device (such as a tape drive), a regular file or a remote tape drive (see rmt(8)). This option must be omitted if the standard output option (a lone - preceding the source filesystem specification) is specified.

-l level

Specifies a dump level of 0 to 9. The dump level determines the base dump to which this dump is relative. The base dump is the most recent dump at a lesser level. A level 0 dump is absolute - all files are dumped. A dump level where $1 \leq \text{level} \leq 9$ is referred to as an incremental dump. Only files that have been changed since the base dump are dumped. Subtree dumps (see the -s option below) cannot be used as the base for incremental dumps.

-m Use the minimal tape protocol for non-scsi tape destinations or remote tape destinations which are not scsi Linux tape drives nor IRIX tape drives. This option cannot be used without specifying a blocksize to be used (see -b option above).

-o Overwrite the tape. With this option, xfsdump does not read the tape first to check the contents. This option may be used if xfs?

dump is unable to determine the block size of a tape .

-p interval

Causes progress reports to be printed at the specified interval.

interval is given in seconds. The progress report indicates how many files have been dumped, the total number of files to dump, the percentage of data dumped, and the elapsed time.

-q Destination tape drive is a QIC tape. QIC tapes only use a 512 byte blocksize, for which xfsdump must make special allowances.

-s pathname [-s pathname ...]

Restricts the dump to files contained in the specified pathnames (subtrees). A pathname must be relative to the mount point of the filesystem.

For example, if a filesystem is mounted at /d2, the pathname argument for the directory /d2/users is ``users". A

pathname can be a file or a directory; if it is a directory, the

entire hierarchy of files and subdirectories rooted at that direc?

tory is dumped. Subtree dumps cannot be used as the base for in? cremental dumps (see the -l option above).

-t file

Sets the dump time to the modification time of file rather than

using the current time. xfsdump uses the dump time to determine

what files need to be backed up during an incremental dump. This

option should be used when dumping snapshots so that the dump time

matches the time the snapshot was taken. Otherwise files modified

after a snapshot is taken may be skipped in the next incremental

dump.

-v verbosity

-v subsystem=verbosity[,subsystem=verbosity,...]

Specifies the level of detail used for messages displayed during

the course of the dump. The verbosity argument can be passed as

either a string or an integer. If passed as a string the following

values may be used: silent, verbose, trace, debug, or nitty. If

passed as an integer, values from 0-5 may be used. The values 0-4

correspond to the strings already listed. The value 5 can be used

to produce even more verbose debug output.

The first form of this option activates message logging across all dump subsystems. The second form allows the message logging level to be controlled on a per-subsystem basis. The two forms can be combined (see the example below). The argument `subsys` can take one of the following values: `general`, `proc`, `drive`, `media`, `inventory`, `inomap` and `excluded_files`.

For example, to dump the root filesystem with tracing activated for all subsystems:

```
# xfsdump -v trace -f /dev/tape /
```

To enable debug-level tracing for drive and media operations:

```
# xfsdump -v drive=debug,media=debug -f /dev/tape /
```

To enable tracing for all subsystems, and debug level tracing for drive operations only:

```
# xfsdump -v trace,drive=debug -f /dev/tape /
```

To list files that will be excluded from the dump:

```
# xfsdump -e -v excluded_files=debug -f /dev/tape /
```

-z size

Specifies the maximum size, in kilobytes, of files to be included in the dump. Files over this size, will be excluded from the dump, except for DMF dual-state files when '-a option' is specified (see '-a option' above). When specified, '-a option' takes precedence over '-z option'. The size is an estimate based on the number of disk blocks actually used by the file, and so does not include holes. In other words, size refers to the amount of space the file would take in the resulting dump. On an interactive restore, the skipped file is visible with `xfsrestore's 'ls'` and while you can use the 'add' and 'extract' commands, nothing will be restored.

-A Do not dump extended file attributes. When dumping a filesystem managed within a DMF environment this option should not be used. DMF stores file migration status within extended attributes associated with each file. If these attributes are not preserved when

the filesystem is restored, files that had been in migrated state will not be recallable by DMF. Note that dumps containing extended file attributes cannot be restored with older versions of xfsre? store(8).

-B session_id

Specifies the ID of the dump session upon which this dump session is to be based. If this option is specified, the **-l** (level) and **-R** (resume) options are not allowed. Instead, xfsdump determines if the current dump session should be incremental and/or resumed, by looking at the base session's level and interrupted attributes. If the base session was interrupted, the current dump session is a resumption of that base at the same level. Otherwise, the current dump session is an incremental dump with a level one greater than that of the base session. This option allows incremental and resumed dumps to be based on any previous dump, rather than just the most recent.

-D Controls which directories are backed up during an incremental dump. By default unchanged directories are dumped if files or directories beneath them have changed. This results in a self-contained dump -- if a base dump is lost, or you know the file(s) you wish to restore is in an incremental dump, you can restore just that dump without loading the base dump(s) first. However, this method requires a potentially expensive traversal through the filesystem.

When **-D** is specified, unchanged directories are not dumped. This results in a faster dump, but files will end up in the xfsre? store(8) orphanage directory unless the base dump(s) is loaded first.

-E Pre-erase media. If this option is specified, media is erased prior to use. The operator is prompted for confirmation, unless the **-F** option is also specified.

-F Don't prompt the operator. When xfsdump encounters a media object containing non-xfsdump data, xfsdump normally asks the operator

for permission to overwrite. With this option the overwrite is performed, no questions asked. When xfsdump encounters end-of-media during a dump, xfsdump normally asks the operator if another media object will be provided. With this option the dump is instead interrupted.

-I Displays the xfsdump inventory (no dump is performed). xfsdump records each dump session in an online inventory in `/var/lib/xfs/dump/inventory`. xfsdump uses this inventory to determine the base for incremental dumps. It is also useful for manually identifying a dump session to be restored. Suboptions to filter the inventory display are described later.

-J Inhibits the normal update of the inventory. This is useful when the media being dumped to will be discarded or overwritten.

-K Generate a format 2 dump instead of the current format. This is useful if the dump will be restored on a system with an older xfsrestore which does not understand the current dump format. Use of this option is otherwise not recommended.

-L session_label

Specifies a label for the dump session. It can be any arbitrary string up to 255 characters long.

-M label [-M label ...]

Specifies a label for the first media object (for example, tape cartridge) written on the corresponding destination during the session. It can be any arbitrary string up to 255 characters long. Multiple media object labels can be specified, one for each destination.

-O options_file

Insert the options contained in options_file into the beginning of the command line. The options are specified just as they would appear if typed into the command line. In addition, newline characters (`\n`) can be used as whitespace. The options are placed before all options actually given on the command line, just after the command name. Only one -O option can be used. Recursive use

is ignored. The source filesystem cannot be specified in options_file.

-R Resumes a previously interrupted dump session. If the most recent dump at this dump's level (-l option) was interrupted, this dump contains only files not in the interrupted dump and consistent with the incremental level. However, files contained in the interrupted dump that have been subsequently modified are re-dumped.

-T Inhibits interactive dialogue timeouts. When the -F option is not specified, xfsdump prompts the operator for labels and media changes. Each dialogue normally times out if no response is supplied. This option prevents the timeout.

-Y length

Specify I/O buffer ring length. xfsdump uses a ring of output buffers to achieve maximum throughput when dumping to tape drives. The default ring length is 3. However, this is not currently enabled on Linux yet, making this option benign.

- A lone - causes the dump stream to be sent to the standard output, where it can be piped to another utility such as xfsrestore(8) or redirected to a file. This option cannot be used with the -f option. The - must follow all other options and precede the filesystem specification.

The filesystem, filesystem, can be specified either as a mount point or as a special device file (for example, /dev/dsk/dks0d1s0). The filesystem must be mounted to be dumped.

NOTES

Dump Interruption

A dump can be interrupted at any time and later resumed. To interrupt, type control-C (or the current terminal interrupt character). The operator is prompted to select one of several operations, including dump interruption. After the operator selects dump interruption, the dump continues until a convenient break point is encountered (typically the end of the current file). Very large files are broken into smaller subfiles, so the wait for the end of the current file is brief.

Dump Resumption

A previously interrupted dump can be resumed by specifying the `-R` option. If the most recent dump at the specified level was interrupted, the new dump does not include files already dumped, unless they have changed since the interrupted dump.

Media Management

A single media object can contain many dump streams. Conversely, a single dump stream can span multiple media objects. If a dump stream is sent to a media object already containing one or more dumps, `xfsdump` appends the new dump stream after the last dump stream. Media files are never overwritten. If end-of-media is encountered during the course of a dump, the operator is prompted to insert a new media object into the drive. The dump stream continuation is appended after the last media file on the new media object.

Inventory

Each dump session updates an inventory database in `/var/lib/xfsdump/inventory`. `xfsdump` uses the inventory to determine the base of incremental and resumed dumps.

This database can be displayed by invoking `xfsdump` with the `-l` option.

The display uses tabbed indentation to present the inventory hierarchically. The first level is filesystem. The second level is session.

The third level is media stream (currently only one stream is supported). The fourth level lists the media files sequentially composing the stream.

The following suboptions are available to filter the display.

`-l depth=n`

(where `n` is 1, 2, or 3) limits the hierarchical depth of the display. When `n` is 1, only the filesystem information from the inventory is displayed. When `n` is 2, only filesystem and session information are displayed. When `n` is 3, only filesystem, session and stream information are displayed.

`-l level=n`

(where `n` is the dump level) limits the display to dumps of that

particular dump level.

The display may be restricted to media files contained in a specific media object.

`-I mobjid=value`

(where value is a media ID) specifies the media object by its media ID.

`-I mobjlabel=value`

(where value is a media label) specifies the media object by its media label.

Similarly, the display can be restricted to a specific filesystem.

`-I mnt=mount_point`

(that is, [hostname:]pathname), identifies the filesystem by mountpoint. Specifying the hostname is optional, but may be useful in a clustered environment where more than one host can be responsible for dumping a filesystem.

`-I fsid=filesystem_id`

identifies the filesystem by filesystem ID.

`-I dev=device_pathname`

(that is, [hostname:]device_pathname) identifies the filesystem by device. As with the mnt filter, specifying the hostname is optional.

More than one of these suboptions, separated by commas, may be specified at the same time to limit the display of the inventory to those dumps of interest. However, at most four suboptions can be specified at once: one to constrain the display hierarchy depth, one to constrain the dump level, one to constrain the media object, and one to constrain the filesystem.

For example, `-I depth=1,mobjlabel="tape 1",mnt=host1:/test_mnt` would display only the filesystem information (depth=1) for those filesystems that were mounted on host1:/test_mnt at the time of the dump, and only those filesystems dumped to the media object labeled "tape 1".

Dump records may be removed (pruned) from the inventory using the `xfstool` program.

An additional media file is placed at the end of each dump stream. This media file contains the inventory information for the current dump session. Its contents may be merged back into the online inventory database at a later time using `xfrestore(1M)`.

The inventory files stored in `/var/lib/xfsdump` are not included in the dump, even if that directory is contained within the filesystem being dumped. Including the inventory in the dump may lead to loss or corruption of data, should an older version be restored overwriting the current version. To backup the `xfsdump` inventory, the contents of `/var/lib/xfsdump` should be copied to another location which may then be safely dumped. Upon restoration, those files may be copied back into `/var/lib/xfsdump`, overwriting whatever files may be there, or `xfsinvt(1M)` may be used to selectively merge parts of the restored inventory back into the current inventory. Prior to version 1.1.8, `xfsdump` would include the `/var/lib/xfsdump` directory in the dump. Care should be taken not to overwrite the `/var/lib/xfsdump` directory when restoring an old dump, by either restoring the filesystem to another location or by copying the current contents of `/var/lib/xfsdump` to a safe place prior to running `xfrestore(1M)`.

Labels

The operator can specify a label to identify the dump session and a label to identify a media object. The session label is placed in every media file produced in the course of the dump, and is recorded in the inventory.

The media label is used to identify media objects, and is independent of the session label. Each media file on the media object contains a copy of the media label. An error is returned if the operator specifies a media label that does not match the media label on a media object containing valid media files. Media labels are recorded in the inventory.

UUIDs

UUIDs (Universally Unique Identifiers) are used in three places: to identify the filesystem being dumped (using the filesystem UUID, see

xfstool(5) for more details), to identify the dump session, and to identify each media object. The inventory display (-l) includes all of these.

Dump Level Usage

The dump level mechanism provides a structured form of incremental dumps. A dump of level *level* includes only files that have changed since the most recent dump at a level less than *level*. For example, the operator can establish a dump schedule that involves a full dump every Friday and a daily incremental dump containing only files that have changed since the previous dump. In this case Friday's dump would be at level 0, Saturday's at level 1, Sunday's at level 2, and so on, up to the Thursday dump at level 6.

The above schedule results in a very tedious restore procedure to fully reconstruct the Thursday version of the filesystem; `xfstool` would need to be fed all 7 dumps in sequence. A compromise schedule is to use level 1 on Saturday, Monday, and Wednesday, and level 2 on Sunday, Tuesday, and Thursday. The Monday and Wednesday dumps would take longer, but the worst case restore requires the accumulation of just three dumps, one each at level 0, level 1, and level 2.

Quotas

If the filesystem being dumped contains user quotas, `xfstool` will use `xfstool_quota(8)` to store the quotas in a file called `xfstool_dump_quotas` in the root of the filesystem to be dumped. This file will then be included in the dump. Upon restoration, `xfstool_quota(8)` can be used to reactivate the quotas for the filesystem. Note, however, that the `xfstool_dump_quotas` file will probably require modification to change the filesystem or UIDs if the filesystem has been restored to a different partition or system. Group and project quotas will be handled in a similar fashion and saved in files called `xfstool_dump_quotas_group` and `xfstool_dump_quotas_proj`, respectively.

Excluding individual files

It may be desirable to exclude particular files or directories from the dump. The `-s` option can be used to limit the dump to a specified directory, and the `-z` option can be used to exclude files over a particu?

lar size. Additionally, when xfsdump is run with the -e option, files that are tagged with the "no dump" file attribute will not be included in the dump. The chattr(1) command can be used to set this attribute on individual files or entire subtrees.

To tag an individual file for exclusion from the dump:

```
$ chattr +d file
```

To tag all files in a subtree for exclusion from the dump:

```
$ chattr -R +d directory
```

Note that any new files or directories created in a directory which has the "no dump" attribute set will automatically inherit this attribute.

Also note that xfsdump does not check directories for the "no dump" attribute.

Care should be taken to note which files have been tagged. Under normal operation, xfsdump will only report the number of files it will skip. The -v excluded_files=debug option, however, will cause xfsdump to list the inode numbers of the individual files affected.

EXAMPLES

To perform a level 0, single stream dump of the root filesystem to a locally mounted tape drive, prompting for session and media labels when required:

```
# xfsdump -f /dev/tape /
```

To specify session and media labels explicitly:

```
# xfsdump -L session_1 -M tape_0 -f /dev/tape /
```

To perform a dump to a remote tape using the minimal rmt protocol and a set blocksize of 64k:

```
# xfsdump -m -b 65536 -f otherhost:/dev/tape /
```

To perform a level 0, multi-stream dump to two locally mounted tape drives:

```
# xfsdump -L session_2 -f /dev/rmt/tps4d6v -M tape_1 \  
-f /dev/rmt/tps5d6v -M tape_2 /
```

To perform a level 1 dump relative to the last level 0 dump recorded in the inventory:

```
# xfsdump -l 1 -f /dev/tape /
```

To copy the contents of a filesystem to another directory (see `xfsrestore(8)`):

```
# xfsdump -J - / | xfsrestore -J - /new
```

FILES

`/var/lib/xfsdump/inventory`

dump inventory database

SEE ALSO

`attr(1)`, `rmt(8)`, `xfsrestore(8)`, `xfsinvutil(8)`, `xfs_quota(8)`,
`attr_get(2)`.

DIAGNOSTICS

The exit code is 0 on normal completion, non-zero if an error occurs or the dump is terminated by the operator.

For all verbosity levels greater than 0 (silent) the final line of the output shows the exit status of the dump. It is of the form:

```
xfsdump: Dump Status: code
```

Where code takes one of the following values: SUCCESS (normal completion), INTERRUPT (interrupted), QUIT (media no longer usable), INCOMPLETE (dump incomplete), FAULT (software error), and ERROR (resource error). Every attempt will be made to keep both the syntax and the semantics of this log message unchanged in future versions of `xfsdump`. However, it may be necessary to refine or expand the set of exit codes, or their interpretation at some point in the future.

The message ```xfsdump: WARNING: unable to open directory: ino N: Invalid argument``` can occur with filesystems which are actively being modified while `xfsdump` is running. This can happen to either directory or regular file inodes - affected files will not end up in the dump, files below affected directories will be placed in the orphanage directory by `xfsrestore`.

BUGS

`xfsdump` does not dump unmounted filesystems.

The dump frequency field of `/etc/fstab` is not supported.

`xfsdump` uses the alert program only when a media change is required.

`xfsdump` requires root privilege (except for inventory display).

xfsdump can only dump XFS filesystems.

The media format used by xfsdump can only be understood by xfsrestore.

xfsdump does not know how to manage CD-ROM or other removable disk drives.

xfsdump can become confused when doing incremental or resumed dumps if on the same machine you dump two XFS filesystems and both filesystems have the same filesystem identifier (UUID). Since xfsdump uses the filesystem identifier to identify filesystems, xfsdump maintains one combined set of dump inventories for both filesystems instead of two sets of dump inventories. This scenario can happen only if dd or some other block-by-block copy program was used to make a copy of an XFS filesystem. See `xfs_copy(8)` and `xfs(5)` for more details.

`xfsdump(8)`