Red Hat Enterprise Linux Release 9.2 Manual Pages on 'xfsrestore.8' command

$ man xfsrestore.8

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

NAME

    xfsrestore - XFS filesystem incremental restore utility

SYNOPSIS

    xfsrestore -h

    xfsrestore [options] -f source [-f source ...] dest

    xfsrestore [options] - dest

    xfsrestore -I [subopt=value ...]

DESCRIPTION

    xfsrestore restores filesystems from dumps produced by xfsdump(8). Two

    modes of operation are available: simple and cumulative.

    The default is simple mode. xfsrestore populates the specified desti?

    nation directory, dest, with the files contained in the dump media.

    The -r option specifies the cumulative mode. Successive invocations of

    xfsrestore are used to apply a chronologically ordered sequence of

    delta dumps to a base (level 0) dump. The contents of the filesystem

    at the time each dump was produced is reproduced.   This can involve

    adding, deleting, renaming, linking, and unlinking files and directo?

    ries.

    A delta dump is defined as either an incremental dump (xfsdump -l op?

    tion with level > 0) or a resumed dump (xfsdump -R option). The deltas

    must be applied in the order they were produced. Each delta applied

    must have been produced with the previously applied delta as its base.

xfsrestore keeps state information in the xfsrestorehousekeepingdir, to inform subsequent invocations when used in cumulative mode, or in the event a restore is interrupted. To ensure that the state information can be processed, a compatible version of xfsrestore must be used for each subsequent invocation. Additionally, each invocation must run on a system of the same endianness and page size.

The options to xfsrestore are:

-a housekeeping

   Each invocation of xfsrestore creates a directory called xfsre?
   storehousekeepingdir. This directory is normally created directly
   under the dest directory. The -a option allows the operator to
   specify an alternate directory, housekeeping, in which xfsrestore
   creates the xfsrestorehousekeepingdir directory. When performing
   a cumulative (-r option) restore or resuming (-R option) a re?
   store, each successive invocation must specify the same alternate
   directory.

-b blocksize

   Specifies the blocksize, in bytes, to be used for the restore.
   For other drives such as DAT or 8 mm , the same blocksize used for
   the xfsdump operation must be specified to restore the tape. The
   default block size is 1Mb.

-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.

-e   Prevents xfsrestore from overwriting existing files in the dest
   directory.

-f source [-f source ...]

   Specifies a source of the dump to be restored. This can be the
   pathname 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 input option (a lone - preceding the dest specifica?
   tion) is specified.

-i   Selects interactive operation. Once the on-media directory hier?

   archy has been read, an interactive dialogue is begun. The opera?

   tor uses a small set of commands to peruse the directory hierar?

   chy, selecting files and subtrees for extraction.   The available

   commands are given below. Initially nothing is selected, except

   for those subtrees specified with -s command line options.

   ls [arg]   List the entries in the current directory or the

            specified directory, or the specified non-directory

            file entry. Both the entry's original inode number

            and name are displayed. Entries that are directo?

            ries are appended with a `/'. Entries that have

            been selected for extraction are prepended with a

            `*'.

   cd [arg]   Change the current working directory to the speci?

            fied argument, or to the filesystem root directory

            if no argument is specified.

   pwd   Print the pathname of the current directory, rela?

            tive to the filesystem root.

   add [arg]   The current directory or specified file or direc?

            tory within the current directory is selected for

            extraction.   If a directory is specified, then it

            and all its descendents are selected. Entries that

            are selected for extraction are prepended with a

            `*' when they are listed by ls.

   delete [arg]   The current directory or specified file or direc?

            tory within the current directory is deselected for

            extraction. If a directory is specified, then it

            and all its descendents are deselected. The most

            expedient way to extract most of the files from a

            directory is to select the directory and then dese?

            lect those files that are not needed.

   extract   Ends the interactive dialogue, and causes all se?

            lected subtrees to be restored.

quit   xfsrestore ends the interactive dialogue and imme?

      diately exits, even if there are files or subtrees

      selected for extraction.

help   List a summary of the available commands.

-m Use the minimal tape protocol. This option cannot be used without

    specifying a blocksize to be used (see -b option above).

-n file

    Allows xfsrestore to restore only files newer than file. The mod?

    ification time of file (i.e., as displayed with the ls -l command)

    is compared to the inode modification time of each file on the

    source media (i.e., as displayed with the ls -lc command). A file

    is restored from media only if its inode modification time is

    greater than or equal to the modification time of file.

-o Restore file and directory owner/group even if not root. When run

    with an effective user id of root, xfsrestore restores owner and

    group of each file and directory. When run with any other effec?

    tive user id it does not, unless this option is specified.

-p interval

    Causes progress reports to be printed at intervals of interval

    seconds. The interval value is approximate, xfsrestore will delay

    progress reports to avoid undue processing overhead.

-q Source tape drive is a QIC tape. QIC tapes only use a 512 byte

    blocksize, for which xfsrestore must make special allowances.

-r Selects the cumulative mode of operation. The -a and destination

    options must be the same for each invocation.

-s subtree

    Specifies a subtree to restore. Any number of -s options are al?

    lowed. The restore is constrained to the union of all subtrees

    specified. Each subtree is specified as a pathname relative to

    the restore dest. If a directory is specified, the directory and

    all files beneath that directory are restored.

-t Displays the contents of the dump, but does not create or modify

    any files or directories. It may be desirable to set the ver?

bosity level to silent when using this option.

-v verbosity

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

> Specifies the level of detail used for messages displayed during
> the course of the restore. 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
> restore 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, inven?
> tory, and tree.
>
> For example, to restore the root filesystem with tracing activated
> for all subsystems:
>
>> # xfsrestore -v trace -f /dev/tape /
>
> To enable debug-level tracing for drive and media operations:
>
>> # xfsrestore -v drive=debug,media=debug -f /dev/tape /
>
> To enable tracing for all subsystems, and debug level tracing for
> drive operations only:
>
>> # xfsrestore -v trace,drive=debug -f /dev/tape /

-A Do not restore extended file attributes. When restoring a
> filesystem managed within a DMF environment this option should not
> be used. DMF stores file migration status within extended at?
> tributes 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 dumping of
> extended file attributes is also optional.

-B Change the ownership and permissions of the destination directory
> to match those of the root directory of the dump.

-D   Restore DMAPI (Data Management Application Programming Interface)

event settings. If the restored filesystem will be managed within

the same DMF environment as the original dump it is essential that

the -D option be used. Otherwise it is not usually desirable to

restore these settings.

-E   Prevents xfsrestore from overwriting newer versions of files. The

inode modification time of the on-media file is compared to the

inode modification time of corresponding file in the dest direc?

tory.   The file is restored only if the on-media version is newer

than the version in the dest directory.   The inode modification

time of a file can be displayed with the ls -lc command.

-F   Inhibit interactive operator prompts. This option inhibits xfsre?

store from prompting the operator for verification of the selected

dump as the restore target and from prompting for any media

change.

-I   Causes the xfsdump inventory to be displayed (no restore is per?

formed).   Each time xfsdump is used, an online inventory in

/var/lib/xfsdump/inventory is updated. This is used to determine

the base for incremental dumps. It is also useful for manually

identifying a dump session to be restored (see the -L and -S op?

tions).   Suboptions to filter the inventory display are described

later.

-J   Inhibits inventory update when on-media session inventory encoun?

tered during restore.   xfsrestore opportunistically updates the

online inventory when it encounters an on-media session inventory,

but only if run with an effective user id of root and only if this

option is not given.

-K   Force xfsrestore to use dump format 2 generation numbers. Normally

the need for this is determined automatically, but this option is

required on the first xfsrestore invocation in the rare case that

a cumulative restore begins with a format 3 (or newer) dump and

will be followed by a format 2 dump.

-L session_label

Specifies the label of the dump session to be restored. The source media is searched for this label. It is any arbitrary string up to 255 characters long. The label of the desired dump session can be copied from the inventory display produced by the -I option.

-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 char? acters (\n) can be used as whitespace. The options are placed be? fore 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 destination directory cannot be specified in op? tions_file.

-Q   Force completion of an interrupted restore session. This option is required to work around one specific pathological scenario. When restoring a dump session which was interrupted due to an EOM condition and no online session inventory is available, xfsrestore cannot know when the restore of that dump session is complete. The operator is forced to interrupt the restore session. In that case, if the operator tries to subsequently apply a resumed dump (using the -r option), xfsrestore refuses to do so. The operator must tell xfsrestore to consider the base restore complete by us? ing this option when applying the resumed dump.

-R   Resume a previously interrupted restore. xfsrestore can be inter? rupted at any time by pressing the terminal interrupt character (see stty(1)). Use this option to resume the restore. The -a and destination options must be the same.

-S session_id

Specifies the session UUID of the dump session to be restored. The source media is searched for this UUID. The UUID of the de? sired dump session can be copied from the inventory display pro? duced by the -I option.

-T   Inhibits interactive dialogue timeouts.   xfsrestore prompts the

   operator for media changes. This dialogue normally times out if

   no response is supplied. This option prevents the timeout.

-X subtree

   Specifies a subtree to exclude. This is the converse of the -s

   option.   Any number of -X options are allowed. Each subtree is

   specified as a pathname relative to the restore dest. If a direc?

   tory is specified, the directory and all files beneath that direc?

   tory are excluded.

-Y io_ring_length

   Specify I/O buffer ring length. xfsrestore uses a ring of input

   buffers to achieve maximum throughput when restoring from tape

   drives. The default ring length is 3. However, this is not cur?

   rently enabled on Linux yet, making this option benign.

-   A lone - causes the standard input to be read as the source of the

   dump to be restored. Standard input can be a pipe from another

   utility (such as xfsdump(8)) or a redirected file. This option

   cannot be used with the -f option. The - must follow all other

   options, and precede the dest specification.

The dumped filesystem is restored into the dest directory. There is no

default; the dest must be specified.

NOTES

 Cumulative Restoration

   A base (level 0) dump and an ordered set of delta dumps can be sequen?

   tially restored, each on top of the previous, to reproduce the contents

   of the original filesystem at the time the last delta was produced.

   The operator invokes xfsrestore once for each dump. The -r option must

   be specified. The dest directory must be the same for all invocations.

   Each invocation leaves a directory named xfsrestorehousekeeping in the

   dest directory (however, see the -a option above). This directory con?

   tains the state information that must be communicated between invoca?

   tions. The operator must remove this directory after the last delta

   has been applied.

xfsrestore also generates a directory named orphanage in the dest di?
rectory. xfsrestore removes this directory after completing a simple
restore.   However, if orphanage is not empty, it is not removed. This
can happen if files present on the dump media are not referenced by any
of the restored directories. The orphanage has an entry for each such
file. The entry name is the file's original inode number, a ".", and
the file's generation count modulo 4096 (only the lower 12 bits of the
generation count are used).

xfsrestore does not remove the orphanage after cumulative restores.
Like the xfsrestorehousekeeping directory, the operator must remove it
after applying all delta dumps.

Media Management

A dump consists of one or more media files contained on one or more me?
dia objects. A media file contains all or a portion of the filesystem
dump. Large filesystems are broken up into multiple media files to
minimize the impact of media dropouts, and to accommodate media object
boundaries (end-of-media).

A media object is any storage medium: a tape cartridge, a remote tape
device (see rmt(8)), a regular file, or the standard input (currently
other removable media drives are not supported). Tape cartridges can
contain multiple media files, which are typically separated by (in tape
parlance) file marks. If a dump spans multiple media objects, the re?
store must begin with the media object containing the first media file
dumped. The operator is prompted when the next media object is needed.
Media objects can contain more than one dump. The operator can select
the desired dump by specifying the dump label (-L option), or by speci?
fying the dump UUID (-S option). If neither is specified, xfsrestore
scans the entire media object, prompting the operator as each dump ses?
sion is encountered.

The inventory display (-I option) is useful for identifying the media
objects required.   It is also useful for identifying a dump session.
The session UUID can be copied from the inventory display to the -S op?
tion argument to unambiguously identify a dump session to be restored.

Dumps placed in regular files or the standard output do not span multi?

ple media objects, nor do they contain multiple dumps.

Inventory

Each dump session updates an inventory database in /var/lib/xfsdump/in?

ventory.   This database can be displayed by invoking xfsrestore with

the -I option. The display uses tabbed indentation to present the in?

ventory 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 sequen?

tially composing the stream.

The following suboptions are available to filter the display.

-I depth=n

   (where n is 1, 2, or 3) limits the hierarchical depth of the dis?

   play. When n is 1, only the filesystem information from the inven?

   tory is displayed. When n is 2, only filesystem and session infor?

   mation are displayed. When n is 3, only filesystem, session and

   stream information are displayed.

-I 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 me?

   dia 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 use?

   ful in a clustered environment where more than one host can be re?

sponsible 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 op?

tional.

More than one of these suboptions, separated by commas, may be speci?

fied 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 xfs?

invutil 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. If the online inventory files in /var/lib/xfsdump/inventory

are missing information for the current dump session, then the inven?

tory information in the media file is automatically added to the files

in /var/lib/xfsdump/inventory.   If you wish to incorporate the inven?

tory information from the media file without restoring any data, you

may do so using the -t option:

xfsrestore -t -f /dev/tape

This is useful to rebuild the inventory database if it is ever lost or

corrupted. The only caveat is that xfsrestore needs to read through

the entire dump in order to reach the inventory media file. This could

become time consuming for dump sessions with large media files.

Media Errors

xfsdump is tolerant of media errors, but cannot do error correction. If a media error occurs in the body of a media file, the filesystem file represented at that point is lost. The bad portion of the media is skipped, and the restoration resumes at the next filesystem file af? ter the bad portion of the media.

If a media error occurs in the beginning of the media file, the entire media file is lost. For this reason, large dumps are broken into a number of reasonably sized media files. The restore resumes with the next media file.

Quotas

When xfsdump dumps a filesystem with user quotas, it creates a file in the root of the dump called xfsdump_quotas. xfsrestore can restore this file like any other file included in the dump. This file can be processed by the restore command of xfs_quota(8) to reactivate the quo? tas. However, the xfsdump_quotas file contains information which may first require modification; specifically the filesystem name and the user ids. If you are restoring the quotas for the same users on the same filesystem from which the dump was taken, then no modification will be necessary. However, if you are restoring the dump to a differ? ent filesystem, you will need to:

- ensure the new filesystem is mounted with the quota option

- modify the xfsdump_quotas file to contain the new filesystem name

- ensure the uids in the xfsdump_quotas file are correct

Once the quota information has been verified, the restore command of xfs_quota (8) can be used to apply the quota limits to the filesystem.

Group and project quotas are handled in a similar fashion and will be restored in files called xfsdump_quotas_group and xfsdump_quotas_proj, respectively.

EXAMPLES

To restore the root filesystem from a locally mounted tape:

    # xfsrestore -f /dev/tape /

To restore from a remote tape, specifying the dump session id:

    # xfsrestore -L session_1 -f otherhost:/dev/tape /new

To restore the contents a of a dump to another subdirectory:

    # xfsrestore -f /dev/tape /newdir

To copy the contents of a filesystem to another directory (see xfs?

dump(8)):

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

FILES

/var/lib/xfsdump/inventory

                    dump inventory database

SEE ALSO

rmt(8), xfsdump(8), xfsinvutil(8), xfs_quota(8), attr_set(2).

DIAGNOSTICS

The exit code is 0 on normal completion, and non-zero if an error oc?

curred or the restore was terminated by the operator.

For all verbosity levels greater than 0 (silent) the final line of the

output shows the exit status of the restore. It is of the form:

    xfsdump: Restore Status: code

Where code takes one of the following values: SUCCESS (normal comple?

tion), INTERRUPT (interrupted), QUIT (media no longer usable), INCOM?

PLETE (restore incomplete), FAULT (software error), and ERROR (resource

error). Every attempt will be made to keep both the syntax and the se?

mantics of this log message unchanged in future versions of xfsrestore.

However, it may be necessary to refine or expand the set of exit codes,

or their interpretation at some point in the future.

BUGS

Pathnames of restored non-directory files (relative to the dest direc?

tory) must be 1023 characters (MAXPATHLEN) or less. Longer pathnames

are discarded and a warning message displayed.

There is no verify option to xfsrestore. This would allow the operator

to compare a filesystem dump to an existing filesystem, without actu?

ally doing a restore.

The interactive commands (-i option) do not understand regular expres?

sions.

When the minimal rmt option is specified, xfsrestore applies it to all

remote tape sources. The same blocksize (specified by the -b option) is used for all these remote drives.

xfsrestore uses the alert program only when a media change is required. Cumulative mode (-r option) requires that the operator invoke xfsre? store for the base and for each delta to be applied in sequence to the base. It would be better to allow the operator to identify the last delta in the sequence of interest, and let xfsrestore work backwards from that delta to identify and apply the preceding deltas and base dump, all in one invocation.

<p align="center">xfsrestore(8)</p>