NAME

btrfs-inspect-internal − query various internal information

SYNOPSIS

btrfs inspect−internal <*subcommand*> <*args*>

DESCRIPTION

This command group provides an interface to query internal information. The functionality ranges from a simple UI to an ioctl or a more complex query that assembles the result from several internal structures. The latter usually requires calls to privileged ioctls.

SUBCOMMAND

dump−super [options] <*device*> [device...]

(replaces the standalone tool **btrfs−show−super**)

Show btrfs superblock information stored on given devices in textual form. By default the first superblock is printed, more details about all copies or additional backup data can be printed.

Besides verification of the filesystem signature, there are no other sanity checks. The superblock checksum status is reported, the device item and filesystem UUIDs are checked and reported.

Note

the meaning of option −*s* has changed in version 4.8 to be consistent with other tools to specify superblock copy rather the offset. The old way still works, but prints a warning. Please update your scripts to use −−*bytenr* instead. The option −*i* has been deprecated.

Options

−f|−−full

print full superblock information, including the system chunk array and backup roots

−a|−−all

print information about all present superblock copies (cannot be used together with −*s* option)

−i <*super*>

(deprecated since 4.8, same behaviour as −−*super*)

−−bytenr <*bytenr*>

specify offset to a superblock in a non−standard location at *bytenr*, useful for debugging (disables the −*f* option)

If there are multiple options specified, only the last one applies.

−F|−−force

attempt to print the superblock even if a valid BTRFS signature is not found; the result may be completely wrong if the data does not resemble a superblock

−s|−−super <*bytenr*>

(see compatibility note above)

specify which mirror to print, valid values are 0, 1 and 2 and the superblock must be present on the device with a valid signature, can be used together with −−*force*

dump−tree [options] <*device*> [device...]

(replaces the standalone tool **btrfs−debug−tree**)

Dump tree structures from a given device in textual form, expand keys to human readable equivalents where possible. This is useful for analyzing filesystem state or inconsistencies and has a positive educational effect on understanding the internal filesystem structure.

Note

contains file names, consider that if you're asked to send the dump for analysis. Does not contain file data.

Options

−e|−−extents
print only extent−related information: extent and device trees

−d|−−device
print only device−related information: tree root, chunk and device trees

−r|−−roots
print only short root node information, ie. the root tree keys

−R|−−backups
same as −−roots plus print backup root info, ie. the backup root keys and the respective tree root block offset

−u|−−uuid
print only the uuid tree information, empty output if the tree does not exist

−b *<block_num>*
print info of the specified block only, can be specified multiple times

−−follow
use with *−b*, print all children tree blocks of *<block_num>*

−−dfs
(default up to 5.2)

use depth−first search to print trees, the nodes and leaves are intermixed in the output −−bfs:::: (default since 5.3)

use breadth−first search to print trees, the nodes are printed before all leaves −−noscan:::: do not automatically scan the system for other devices from the same filesystem, only use the devices provided as the arguments −t *<tree_id>*:::: print only the tree with the specified ID, where the ID can be numerical or common name in a flexible human readable form

The tree id name recognition rules:

- case does not matter
- the C source definition, eg. BTRFS_ROOT_TREE_OBJECTID
- short forms without BTRFS_ prefix, without _TREE and _OBJECTID suffix, eg. ROOT_TREE, ROOT
- convenience aliases, eg. DEVICE for the DEV tree, CHECKSUM for CSUM
- unrecognized ID is an error

inode−resolve [−v] *<ino>* *<path>*
(needs root privileges)

resolve paths to all files with given inode number *ino* in a given subvolume at *path*, ie. all hardlinks

Options

−v
verbose mode, print count of returned paths and ioctl() return value

logical−resolve [−Pv] [−s *<bufsize>*] *<logical>* *<path>*
(needs root privileges)

resolve paths to all files at given *logical* address in the linear filesystem space

Options

−P
: skip the path resolving and print the inodes instead

−v
: verbose mode, print count of returned paths and all ioctl() return values

−s <*bufsize*>
: set internal buffer for storing the file names to *bufsize*, default is 4096, maximum 64k

min−dev−size [options] <*path*>

(needs root privileges)

return the minimum size the device can be shrunk to, without performing any resize operation, this may be useful before executing the actual resize operation

Options

−−id <*id*>
: specify the device *id* to query, default is 1 if this option is not used

rootid <*path*>

for a given file or directory, return the containing tree root id, but for a subvolume itself return its own tree id (ie. subvol id)

Note

The result is undefined for the so−called empty subvolumes (identified by inode number 2), but such a subvolume does not contain any files anyway

subvolid−resolve <*subvolid*> <*path*>

(needs root privileges)

resolve the absolute path of the subvolume id *subvolid*

tree−stats [options] <*device*>

(needs root privileges)

Print sizes and statistics of trees.

Options

−b
: Print raw numbers in bytes.

EXIT STATUS

btrfs inspect−internal returns a zero exit status if it succeeds. Non zero is returned in case of failure.

AVAILABILITY

btrfs is part of btrfs−progs. Please refer to the btrfs wiki **http://btrfs.wiki.kernel.org** for further details.

SEE ALSO

mkfs.btrfs(8)