NAME

btrfs-check − check or repair a btrfs filesystem

SYNOPSIS

btrfs check [options] <*device*>

DESCRIPTION

The filesystem checker is used to verify structural integrity of a filesystem and attempt to repair it if requested. It is recommended to unmount the filesystem prior to running the check, but it is possible to start checking a mounted filesystem (see −−*force*).

By default, **btrfs check** will not modify the device but you can reaffirm that by the option −−*readonly*.

btrfsck is an alias of **btrfs check** command and is now deprecated.

Warning

Do not use −−*repair* unless you are advised to do so by a developer or an experienced user, and then only after having accepted that no *fsck* successfully repair all types of filesystem corruption. Eg. some other software or hardware bugs can fatally damage a volume.

The structural integrity check verifies if internal filesystem objects or data structures satisfy the constraints, point to the right objects or are correctly connected together.

There are several cross checks that can detect wrong reference counts of shared extents, backreferences, missing extents of inodes, directory and inode connectivity etc.

The amount of memory required can be high, depending on the size of the filesystem, similarly the run time. Check the modes that can also affect that.

SAFE OR ADVISORY OPTIONS

−b|−−backup
: use the first valid set of backup roots stored in the superblock

  This can be combined with −−*super* if some of the superblocks are damaged.

−−check−data−csum
: verify checksums of data blocks

  This expects that the filesystem is otherwise OK, and is basically an offline *scrub* that does not repair data from spare copies.

−−chunk−root <*bytenr*>
: use the given offset *bytenr* for the chunk tree root

−E|−−subvol−extents <*subvolid*>
: show extent state for the given subvolume

−p|−−progress
: indicate progress at various checking phases

−Q|−−qgroup−report
: verify qgroup accounting and compare against filesystem accounting

−r|−−tree−root <*bytenr*>
: use the given offset *bytenr* for the tree root

−−readonly
: (default) run in read−only mode, this option exists to calm potential panic when users are going to run the checker

−s|−−super <*superblock*>
: use 'superblock’th superblock copy, valid values are 0, 1 or 2 if the respective superblock offset is

within the device size

This can be used to use a different starting point if some of the primary superblock is damaged.

−−clear−space−cache v1|v2
completely wipe all free space cache of given type

For free space cache *v1*, the *clear_cache* kernel mount option only rebuilds the free space cache for block groups that are modified while the filesystem is mounted with that option. Thus, using this option with *v1* makes it possible to actually clear the entire free space cache.

For free space cache *v2*, the *clear_cache* kernel mount option destroys the entire free space cache. This option, with *v2* provides an alternative method of clearing the free space cache that doesn't require mounting the filesystem.

−−clear−ino−cache
remove leftover items pertaining to the deprecated inode map feature

DANGEROUS OPTIONS

−−repair
enable the repair mode and attempt to fix problems where possible

Note
there's a warning and 10 second delay when this option is run without −−*force* to give users a chance to think twice before running repair, the warnings in documentation have shown to be insufficient

−−init−csum−tree
create a new checksum tree and recalculate checksums in all files

Note
Do not blindly use this option to fix checksum mismatch problems.

−−init−extent−tree
build the extent tree from scratch

Note
Do not use unless you know what you're doing.

−−mode <*MODE*>
select mode of operation regarding memory and IO

The *MODE* can be one of:

original
The metadata are read into memory and verified, thus the requirements are high on large filesystems and can even lead to out−of−memory conditions. The possible workaround is to export the block device over network to a machine with enough memory.

lowmem
This mode is supposed to address the high memory consumption at the cost of increased IO when it needs to re−read blocks. This may increase run time.

Note
lowmem mode does not work with −−*repair* yet, and is still considered experimental.

−−force
allow work on a mounted filesystem. Note that this should work fine on a quiescent or read−only mounted filesystem but may crash if the device is changed externally, eg. by the kernel module. Repair without mount checks is not supported right now.

This option also skips the delay and warning in the repair mode (see −−*repair*).

EXIT STATUS

btrfs check 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), **btrfs−scrub**(8), **btrfs−rescue**(8)