

NAME

btrfstune – tune various filesystem parameters

SYNOPSIS

btrfstune [options] <device> [<device>...]

DESCRIPTION

btrfstune can be used to enable, disable, or set various filesystem parameters. The filesystem must be unmounted.

The common use case is to enable features that were not enabled at mkfs time. Please make sure that you have kernel support for the features. You can find a complete list of features and kernel version of their introduction at Feature by version <> page. Also, the manual page mkfs.btrfs(8) <> contains more details about the features.

Some of the features could be also enabled on a mounted filesystem by other means. Please refer to the *FILESYSTEM FEATURES* in btrfs(5) <>.

Note:

btrfstune requires the target filesystem to have no running replace nor balance. Please finish/cancel the running replace/balance before using **btrfstune**.

OPTIONS

--convert-to-block-group-tree
(since kernel 6.1)

Convert portions of extent tree that tracks block groups to a separate block group tree. This greatly reduces mount time. Can be also enabled at mkfs time.

--convert-from-block-group-tree
(since kernel 6.1)

Convert block groups tracked in standalone block group tree back to extent tree and remove *block-group-tree* feature bit from the filesystem.

--convert-to-free-space-tree
(since kernel 4.5)

Convert to free-space-tree feature (v2 of space cache).

--enable-simple-quota
(since kernel 6.7)

Enable simple quota accounting (*squota* mkfs feature). This is an alternative to qgroups with a smaller performance impact but no notion of shared vs. exclusive usage.

--remove-simple-quota

Remove simple quota accounting related structures.

-f Allow dangerous changes, e.g. clear the seeding flag or change fsid. Make sure that you are aware of the dangers.

-m (since kernel: 5.0)

change fsid stored as *metadata_uuid* to a randomly generated UUID, see also *-U*

-M <UUID>

(since kernel: 5.0)

change fsid stored as *metadata_uuid* to a given UUID, see also *-U*

The `metadata_uuid` is stored only in the superblock and is a backward incompatible change. The `fsid` in metadata blocks remains unchanged and is not overwritten, thus the whole operation is significantly faster than `-U`.

The new `metadata_uuid` can be used for mount by UUID and is also used to identify devices of a multi-device filesystem.

-n (since kernel: 3.14)

Enable no-holes feature (more efficient representation of file holes), enabled by mkfs feature *no-holes*.

-r (since kernel: 3.7)

Enable extended inode refs (hardlink limit per file in a directory is 65536), enabled by mkfs feature *extref*.

-S <0|1>

Enable seeding on a given device. Value 1 will enable seeding, 0 will disable it. A seeding filesystem is forced to be mounted read-only. A new device can be added to the filesystem and will capture all writes keeping the seeding device intact. See also section SEEDING DEVICE <#man-btrfs5-seeding-device> in `btrfs(5)` <>.

Warning:

Clearing the seeding flag on a device may be dangerous. If a previously-seeding device is changed, all filesystems that used that device will become unmountable. Setting the seeding flag back will not fix that.

A valid usecase is *seeding device as a base image*. Clear the seeding flag, update the filesystem and make it seeding again, provided that it's OK to throw away all filesystems built on top of the previous base.

-u Change `fsid` to a randomly generated UUID or continue previous `fsid` change operation in case it was interrupted.

-U <UUID>

Change `fsid` to *UUID* in all metadata blocks.

The *UUID* should be a 36 bytes string in printf(3) <<https://man7.org/linux/man-pages/man3/printf.3.html>> format `%08x-%04x-%04x-%04x-%012x`. If there is a previous unfinished `fsid` change, it will continue only if the *UUID* matches the unfinished one or if you use the option `-u`.

All metadata blocks are rewritten, this may take some time, but the final filesystem compatibility is unaffected, unlike `-M`.

Warning:

Cancelling or interrupting a UUID change operation will make the filesystem temporarily unmountable. To fix it, rerun **btrfs(8) -u** and let it complete.

-x (since kernel: 3.10)

Enable skinny metadata extent refs (more efficient representation of extents), enabled by mkfs feature *skinny-metadata*.

All newly created extents will use the new representation. To completely switch the entire filesystem, run a full balance of the metadata. Please refer to `btrfs-balance(8)` <>.

--version

Print the **btrfs** version, builtin features and exit.

EXIT STATUS

btrfs returns 0 if no error happened, 1 otherwise.

COMPATIBILITY NOTE

This deprecated tool exists for historical reasons but is still in use today. Its functionality will be merged to the main tool, at which time **btrfs** will be declared obsolete and scheduled for removal.

SEE ALSO

btrfs(5) <>, btrfs-balance(8) <>, mkfs.btrfs(8) <>