NAME

btrfs-convert − convert from ext2/3/4 or reiserfs filesystem to btrfs in−place

SYNOPSIS

btrfs−convert [options] <*device*>

DESCRIPTION

btrfs−convert is used to convert existing source filesystem image to a btrfs filesystem in−place. The original filesystem image is accessible in subvolume named like *ext2_saved* as file *image*.

Supported filesystems:

- ext2, ext3, ext4 — original feature, always built in
- reiserfs — since version 4.13, optionally built, requires libreiserfscore 3.6.27
- ntfs — external tool **https://github.com/maharmstone/ntfs2btrfs**

The list of supported source filesystem by a given binary is listed at the end of help (option −−*help*).

Warning

If you are going to perform rollback to the original filesystem, you should not execute **btrfs balance** command on the converted filesystem. This will change the extent layout and make **btrfs−convert** unable to rollback.

The conversion utilizes free space of the original filesystem. The exact estimate of the required space cannot be foretold. The final btrfs metadata might occupy several gigabytes on a hundreds−gigabyte filesystem.

If the ability to rollback is no longer important, the it is recommended to perform a few more steps to transition the btrfs filesystem to a more compact layout. This is because the conversion inherits the original data blocks' fragmentation, and also because the metadata blocks are bound to the original free space layout.

Due to different constraints, it is only possible to convert filesystems that have a supported data block size (ie. the same that would be valid for *mkfs.btrfs*). This is typically the system page size (4KiB on x86_64 machines).

BEFORE YOU START

The source filesystem must be clean, eg. no journal to replay or no repairs needed. The respective *fsck* utility must be run on the source filesystem prior to conversion. Please refer to the manual pages in case you encounter problems.

For ext2/3/4:

e2fsck −fvy /dev/sdx

For reiserfs:

reiserfsck −fy /dev/sdx

Skipping that step could lead to incorrect results on the target filesystem, but it may work.

REMOVE THE ORIGINAL FILESYSTEM METADATA

By removing the subvolume named like *ext2_saved* or *reiserfs_saved*, all metadata of the original filesystem will be removed:

btrfs subvolume delete /mnt/ext2_saved

At this point it is not possible to do a rollback. The filesystem is usable but may be impacted by the fragmentation inherited from the original filesystem.

MAKE FILE DATA MORE CONTIGUOUS

An optional but recommended step is to run defragmentation on the entire filesystem. This will attempt to make file extents more contiguous.

btrfs filesystem defrag −v −r −f −t 32M /mnt/btrfs

Verbose recursive defragmentation (*−v*, *−r*), flush data per−file (*−f*) with target extent size 32MiB (*−t*).

ATTEMPT TO MAKE BTRFS METADATA MORE COMPACT

Optional but recommended step.

The metadata block groups after conversion may be smaller than the default size (256MiB or 1GiB). Running a balance will attempt to merge the block groups. This depends on the free space layout (and fragmentation) and may fail due to lack of enough work space. This is a soft error leaving the filesystem usable but the block group layout may remain unchanged.

Note that balance operation takes a lot of time, please see also **btrfs−balance**(8).

btrfs balance start −m /mnt/btrfs

OPTIONS

−−csum <*type*>, −−checksum <*type*>
: Specify the checksum algorithm. Default is *crc32c*. Valid values are *crc32c*, *xxhash*, *sha256* or *blake2*. To mount such filesystem kernel must support the checksums as well.

−d|−−no−datasum
: disable data checksum calculations and set the NODATASUM file flag, this can speed up the conversion

−i|−−no−xattr
: ignore xattrs and ACLs of files

−n|−−no−inline
: disable inlining of small files to metadata blocks, this will decrease the metadata consumption and may help to convert a filesystem with low free space

−N|−−nodesize <*SIZE*>
: set filesystem nodesize, the tree block size in which btrfs stores its metadata. The default value is 16KB (16384) or the page size, whichever is bigger. Must be a multiple of the sectorsize, but not larger than 65536. See **mkfs.btrfs**(8) for more details.

−r|−−rollback
: rollback to the original ext2/3/4 filesystem if possible

−l|−−label <*LABEL*>
: set filesystem label during conversion

−L|−−copy−label
: use label from the converted filesystem

−O|−−features <*feature1*>[,<*feature2*>...]
: A list of filesystem features enabled the at time of conversion. Not all features are supported by old kernels. To disable a feature, prefix it with ^. Description of the features is in section *FILESYSTEM FEATURES* of **mkfs.btrfs**(8).

  To see all available features that btrfs−convert supports run:

  btrfs−convert −O list−all

−p|−−progress
show progress of conversion (a heartbeat indicator and number of inodes processed), on by default

−−no−progress
disable progress and show only the main phases of conversion

−−uuid <*SPEC*>
set the FSID of the new filesystem based on *SPEC*:

- *new* − (default) generate UUID for the FSID of btrfs
- *copy* − copy UUID from the source filesystem
- *UUID* − a conforming UUID value, the 36 byte string representation

EXIT STATUS

btrfs−convert will return 0 if no error happened. If any problems happened, 1 will be returned.

SEE ALSO

mkfs.btrfs(8)