OVERVIEW
Introduction
Status
Manual pages
btrfs(8)
btrfs(5)
btrfs-balance(8)
btrfs-check(8)
btrfs-convert(8)
SYNOPSIS
DESCRIPTION
OPTIONS
EXIT STATUS
SEE ALSO
btrfs-device(8)
btrfs-filesystem(8)
btrfs-find-root(8)
btrfs-image(8)
btrfs-inspect-internal(8)
btrfs-ioctl(2)
btrfs-map-logical(8)
btrfs-property(8)
btrfs-qgroup(8)
btrfs-quota(8)
btrfs-receive(8)
btrfs-replace(8)
btrfs-rescue(8)
btrfs-restore(8)
btrfs-scrub(8)
btrfs-select-super(8)
btrfs-send(8)
btrfs-subvolume(8)
btrfstune(8)
fsck.btrfs(8)
mkfs.btrfs(8)
Administration
Hardware considerations
Changes (feature/version)
Changes (kernel/version)
Changes (btrfs-progs)
Contributors
Glossary
Installation instructions
Source repositories
Interoperability
FEATURES
Common Linux features
Custom ioctls
Auto-repair on read
Balance
Compression
Checksumming
Convert
Deduplication
Defragmentation
Inline files
Quota groups
Reflink
Resize
Scrub
Seeding device
Send/receive
Subpage support
Subvolumes
Swapfile
Tree checker
Trim/discard
Volume management
Zoned mode
DEVELOPER DOCUMENTATION
Development notes
Developer's FAQ
Conventions and style for documentation
Experimental features
Btrfs design
Btrees
On-disk Format
Send stream format
JSON output
Internal APIs
Release checklist
Pull request review workflow
Command line, formatting, UI guidelines
btrfs-ioctl(2)
TODO
Troubleshooting pages



# btrfs-convert(8)

## SYNOPSIS

**btrfs-convert** [options] <device>

## DESCRIPTION

The **btrfs-convert** tool can be 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 (i.e. the same that would be valid for **mkfs.btrfs**). This is typically the system page size (4KiB on x86_64 machines).

**Note**

Always consider if a mkfs and file copy would not be a better option than the in-place conversion given what was said above. The conversion depends on 3rd party libraries and the other filesystems could still evolve and add new features. Not all combinations are covered or tested.

### BEFORE YOU START

The source filesystem must be clean, e.g. 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 16KiB (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**

OVERVIEW
Introduction
Status
Manual pages
btrfs(8)
btrfs(5)
btrfs-balance(8)
btrfs-check(8)
btrfs-convert(8)
SYNOPSIS
DESCRIPTION
OPTIONS
EXIT STATUS
SEE ALSO
btrfs-device(8)
btrfs-filesystem(8)
btrfs-find-root(8)
btrfs-image(8)
btrfs-inspect-internal(8)
btrfs-ioctl(2)
btrfs-map-logical(8)
btrfs-property(8)
btrfs-qgroup(8)
btrfs-quota(8)
btrfs-receive(8)
btrfs-replace(8)
btrfs-rescue(8)
btrfs-restore(8)
btrfs-scrub(8)
btrfs-select-super(8)
btrfs-send(8)
btrfs-subvolume(8)
btrfstune(8)
fsck.btrfs(8)
mkfs.btrfs(8)
Administration
Hardware considerations
Changes (feature/version)
Changes (kernel/version)
Changes (btrfs-progs)
Contributors
Glossary
Installation instructions
Source repositories
Interoperability
FEATURES
Common Linux features
Custom ioctls
Auto-repair on read
Balance
Compression
Checksumming
Convert
Deduplication
Defragmentation
Inline files
Quota groups
Reflink
Resize
Scrub
Seeding device
Send/receive
Subpage support
Subvolumes
Swapfile
Tree checker
Trim/discard
Volume management
Zoned mode
DEVELOPER DOCUMENTATION
Development notes
Developer's FAQ
Conventions and style for documentation
Experimental features
Btrfs design
Btrees
On-disk Format
Send stream format
JSON output
Internal APIs
Release checklist
Pull request review workflow
Command line, formatting, UI guidelines
btrfs-ioctl(2)
TODO
Troubleshooting pages

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

**`--version`**

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

## EXIT STATUS

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

## SEE ALSO

mkfs.btrfs(8)



© Copyright .

Built with Sphinx using a theme provided by Read the Docs.