BTRFS
stable

Search docs

OVERVIEW
Introduction
Status
Manual pages
btrfs(8)
btrfs(5)
btrfs-balance(8)
btrfs-check(8)
btrfs-convert(8)
btrfs-device(8)
btrfs-filesystem(8)
btrfs-find-root(8)
btrfs-image(8)
btrfs-inspect-internal(8)
SYNOPSIS
DESCRIPTION
SUBCOMMAND
EXIT STATUS
AVAILABILITY
SEE ALSO
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

btrfs-inspect-internal(8)

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...]

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...]

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**
>
> By default contains file names, consider that if you're asked to send the dump for analysis and use *--hide-names* eventually. Does not contain file data.

Special characters in file names, xattr names and values are escaped, in the C style like `\n` and octal encoding `\NNN`.

`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, i.e. the root tree keys

-R|--backups

same as *--roots* plus print backup root info, i.e. 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

`--hide-names`

print a placeholder *HIDDEN* instead of various names, useful for developers to inspect the dump while keeping potentially sensitive information hidden

This is:

- directory entries (files, directories, subvolumes)
- default subvolume
- extended attributes (name, value)
- hardlink names (if stored inside another item or as extended references in standalone items)

> **Note**
>
> Lengths are not hidden because they can be calculated from the item size anyway.

`--csum-headers`

print b-tree node checksums stored in headers (metadata)

stable

BTRFS

stable

Search docs

OVERVIEW

Introduction

Status

Manual pages

btrfs(8)

btrfs(5)

btrfs-balance(8)

btrfs-check(8)

btrfs-convert(8)

btrfs-device(8)

btrfs-filesystem(8)

btrfs-find-root(8)

btrfs-image(8)

btrfs-inspect-internal(8)

SYNOPSIS

DESCRIPTION

SUBCOMMAND

EXIT STATUS

AVAILABILITY

SEE ALSO

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

`--csum-items`

print checksums stored in checksum items (data)

`--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, e.g. BTRFS_ROOT_TREE_OBJECTID
- short forms without BTRFS_ prefix, without _TREE and _OBJECTID suffix, e.g. ROOT_TREE, ROOT
- convenience aliases, e.g. 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*, i.e. all hardlinks

`Options`

`-v`

(deprecated) alias for global -v option

logical-resolve [-Pvo] [-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

`-o`

ignore offsets, find all references to an extent instead of a single block. Requires kernel support for the V2 ioctl (added in 4.15). The results might need further processing to filter out unwanted extents by the offset that is supposed to be obtained by other means.

`-s <bufsize>`

set internal buffer for storing the file names to *bufsize*, default is 64KiB, maximum 16MiB. Buffer sizes over 64Kib require kernel support for the V2 ioctl (added in 4.15).

`-v`

(deprecated) alias for global -v option

list-chunks [options] <path>

(needs root privileges)

Enumerate chunks on all devices. The chunks represent the physical range on devices (not to be confused with block groups that represent the logical ranges, but the terms are often used interchangeably).

Example output:

```
Devid PNumber      Type/profile    PStart    Length      PEnd LNumber    LStart Usage%
----- ------- ----------------- --------- --------- --------- ------- --------- ------
    1       1       Data/single   1.00MiB  84.00MiB  85.00MiB      68 191.60GiB  62.77
    1       2     System/DUP     85.00MiB  32.00MiB 117.00MiB      39 140.17GiB   0.05
    1       3     System/DUP    117.00MiB  32.00MiB 149.00MiB      40 140.17GiB   0.05
    1       4   Metadata/DUP    149.00MiB 192.00MiB 341.00MiB      59 188.41GiB  45.00
    1       5   Metadata/DUP    341.00MiB 192.00MiB 533.00MiB      60 188.41GiB  45.00
    1       6       Data/single 533.00MiB   1.00GiB   1.52GiB      49 169.91GiB  72.23
    1       7       Data/single   1.52GiB  16.00MiB   1.54GiB      69 191.68GiB  79.83
    1       8       Data/single   1.54GiB   1.00GiB   2.54GiB      17 100.90GiB  46.39
    1       9       Data/single   2.54GiB   1.00GiB   3.54GiB      16  99.90GiB  40.68
    1      10       Data/single   3.54GiB   1.00GiB   4.54GiB       1  71.40GiB  62.97
    1      11       Data/single   4.54GiB   1.00GiB   5.54GiB      33 125.04GiB  26.00
    1      12       Data/single   5.54GiB   1.00GiB   6.54GiB      50 170.91GiB  60.44
    1      13       Data/single   6.54GiB 512.00MiB   7.04GiB      63 189.16GiB  67.34
    1      14       Data/single   7.04GiB   1.00GiB   8.04GiB      51 171.91GiB  70.94
```

- *Devid* -- the device id
- *PNumber* -- the number of the chunk on the device (in order)
- *Type/profile* -- the chunk type and profile
- *PStart* -- the chunk start on the device
- *Length* -- the chunk length (same for physical and logical address space)
- *PEnd* -- the chunk end, effectively *PStart* + *Length*
- *LNumber* -- the number of the chunk, in the logical address space of the whole filesystem
- *LStart* -- the chunk start in the logical address space of the whole filesystem, as it's a single space it's also called *offset*
- *Usage* -- chunk usage, percentage of used data/metadata of the chunk length

The chunks in the output can be sorted by one or more sorting criteria, evaluated as specified, in the ascending order. By default the chunks are sorted by *devid* and *pstart*, this is most convenient for single device filesystems.

On multi-device filesystems it's up to the user what is preferred as the layout of chunks on e.g. striped profiles (RAID0 etc) cannot be easily represented. A logical view with corresponding underlying structure would be better, but sorting by *lstart,devid* at least groups devices of the given logical range. Can be also combined with *usage*.

This output can provide information for balance filters.

`Options`

`--sort MODE`

sort by a column (ascending):

MODE is a coma separated list of:

devid - by device id (default, with pstart)

pstart - physical start (relative to the beginning of the device)

lstart - logical offset (in the logical address space)

usage - by chunk usage (percentage)

length - by chunk length

`--raw`

raw numbers in bytes, without the *B* suffix

`--human-readable`

print human friendly numbers, base 1024, this is the default

`--iec`

select the 1024 base for the following options, according to the IEC standard

`--si`

select the 1000 base for the following options, according to the SI standard

`--kbytes`

show sizes in KiB, or kB with --si

`--mbytes`

show sizes in MiB, or MB with --si

BTRFS
stable

Search docs

OVERVIEW
Introduction
Status
Manual pages
btrfs(8)
btrfs(5)
btrfs-balance(8)
btrfs-check(8)
btrfs-convert(8)
btrfs-device(8)
btrfs-filesystem(8)
btrfs-find-root(8)
btrfs-image(8)
btrfs-inspect-internal(8)
SYNOPSIS
DESCRIPTION
SUBCOMMAND
EXIT STATUS
AVAILABILITY
SEE ALSO
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

`--gbytes`

show sizes in GiB, or GB with --si

`--tbytes`

show sizes in TiB, or TB with --si

map-swapfile [options] <file>

(needs root privileges)

Find device-specific physical offset of *file* that can be used for hibernation. Also verify that the *file* is suitable as a swapfile. See also command **btrfs filesystem mkswapfile** and the Swapfile feature description.

> **Note**
>
> Do not use **filefrag** or *FIEMAP* ioctl values reported as physical, this is different due to internal filesystem mappings. The hibernation expects offset relative to the physical block device.

`Options`

-r|--resume-offset

print only the value suitable as resume offset for file */sys/power/resume_offset*

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 (i.e. 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. This takes a device as an argument and not a mount point unlike other commands.

> **Note**
>
> In case the the filesystem is still mounted it's possible to run the command but the results may be inaccurate or various errors may be printed in case there are ongoing writes to the filesystem. A warning is printed in such case.

`Options`

-b|--raw

raw numbers in bytes, without the *B* suffix

`-t` *`<treeid>`*

Print stats only for the given treeid.

`--human-readable`

print human friendly numbers, base 1024, this is the default

`--iec`

select the 1024 base for the following options, according to the IEC standard

`--si`

select the 1000 base for the following options, according to the SI standard

`--kbytes`

show sizes in KiB, or kB with --si

`--mbytes`

show sizes in MiB, or MB with --si

`--gbytes`

show sizes in GiB, or GB with --si

`--tbytes`

show sizes in TiB, or TB with --si

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 documentation at https://btrfs.readthedocs.io.

SEE ALSO

mkfs.btrfs(8)

Previous Next

stable