



Linux Ubuntu 26.04 Manual Pages on command 'chattr.1'

\$ man chattr.1

CHATTR(1) General Commands Manual CHATTR(1)

NAME

chattr - change file attributes on a Linux file system

SYNOPSIS

chattr [-RVf] [-v version] [-p project] [mode] files...

DESCRIPTION

chattr changes the file attributes on a Linux file system.

The format of a symbolic mode is +=[aAcCdDeFijmPsStTux].

The operator '+' causes the selected attributes to be added to the existing attributes of the files; '-' causes them to be removed; and '=' causes them to be the only attributes that the files have.

The letters 'aAcCdDeFijmPsStTux' select the new attributes for the files: append only (a), no atime updates (A), compressed (c), no copy on write (C), no dump (d), synchronous directory updates (D), extent format (e), case-insensitive directory lookups (F), immutable (i), data journaling (j), don't compress (m), project hierarchy (P), secure deletion (s), synchronous updates (S), no tail-merging (t), top of directory hierarchy (T), undeletable (u), and direct access for files (x).

The following attributes are read-only, and may be listed by lsattr(1) but not modified by chattr: encrypted (E), indexed directory (I), inline data (N), and verity (V).

Not all flags are supported or utilized by all file systems; refer to file system-specific man pages such as btrfs(5), ext4(5), mkfs.f2fs(8), and xfs(5) for more file system-specific details.

OPTIONS

-R Recursively change attributes of directories and their contents.

-V Be verbose with chattr's output and print the program version.

-f Suppress most error messages.

-v version

Set the file's version/generation number.

-p project

Set the file's project number.

ATTRIBUTES

a A file with the 'a' attribute set can only be opened in append mode for writing.

Only the superuser or a process possessing the CAP_LINUX_IMMUTABLE capability can set or clear this attribute.

A When a file with the 'A' attribute set is accessed, its atime record is not modified.

This avoids a certain amount of disk I/O for laptop systems.

c A file with the 'c' attribute set is automatically compressed on the disk by the kernel.

A read from this file returns uncompressed data. A write to this file compresses data before storing them on the disk. Note: please make sure to read the bugs and limitations section at the end of this document. (Note: For btrfs, if the 'c' flag is set, then the 'C' flag cannot be set. Also conflicts with btrfs mount option 'nodatasum')

C A file with the 'C' attribute set will not be subject to copy-on-write updates. This flag is only supported on file systems which perform copy-on-write. (Note: For btrfs, the 'C' flag should be set on new or empty files. If it is set on a file which already has data blocks, it is undefined when the blocks assigned to the file will be fully stable. If the 'C' flag is set on a directory, it will have no effect on the directory, but new files created in that directory will have the No_COW attribute set. If the 'C' flag is set, then the 'c' flag cannot be set.)

d A file with the 'd' attribute set is not a candidate for backup when the dump(8) program is run.

D When a directory with the 'D' attribute set is modified, the changes are written synchronously to the disk; this is equivalent to the 'dirdsync' mount option applied to a subset of the files.

e The 'e' attribute indicates that the file is using extents for mapping the blocks on disk. It may not be removed using chattr(1).

- E A file, directory, or symlink with the 'E' attribute set is encrypted by the file system. This attribute may not be set or cleared using `chattr(1)`, although it can be displayed by `lsattr(1)`.
- F A directory with the 'F' attribute set indicates that all the path lookups inside that directory are made in a case-insensitive fashion. This attribute can only be changed in empty directories on file systems with the `casefold` feature enabled.
- i A file with the 'i' attribute cannot be modified: it cannot be deleted or renamed, no link can be created to this file, most of the file's metadata can not be modified, and the file can not be opened in write mode. Only the superuser or a process possessing the `CAP_LINUX_IMMUTABLE` capability can set or clear this attribute.
- I The 'I' attribute is used by the `htree` code to indicate that a directory is being indexed using hashed trees. It may not be set or cleared using `chattr(1)`, although it can be displayed by `lsattr(1)`.
- j A file with the 'j' attribute has all of its data written to the `ext3` or `ext4` journal before being written to the file itself, if the file system is mounted with the `"data=ordered"` or `"data=writeback"` options and the file system has a journal. When the file system is mounted with the `"data=journal"` option all file data is already journalled and this attribute has no effect. Only the superuser or a process possessing the `CAP_SYS_RESOURCE` capability can set or clear this attribute.
- m A file with the 'm' attribute is excluded from compression on file systems that support per-file compression.
- N A file with the 'N' attribute set indicates that the file has data stored inline, within the inode itself. It may not be set or cleared using `chattr(1)`, although it can be displayed by `lsattr(1)`.
- P A directory with the 'P' attribute set will enforce a hierarchical structure for project id's. This means that files and directories created in the directory will inherit the project id of the directory, rename operations are constrained so when a file or directory is moved into another directory, that the project ids must match. In addition, a hard link to file can only be created when the project id for the file and the destination directory match.
- s When a file with the 's' attribute set is deleted, its blocks are zeroed and written back to the disk. Note: please make sure to read the bugs and limitations section at the end of this document.

- S When a file with the 'S' attribute set is modified, the changes are written synchronously to the disk; this is equivalent to the 'sync' mount option applied to a subset of the files.
 - t A file with the 't' attribute will not have a partial block fragment at the end of the file merged with other files (for those file systems which support tail-merging). This is necessary for applications such as LILO which read the file system directly, and which don't understand tail-merged files. Note: As of this writing, the ext2, ext3, and ext4 file systems do not support tail-merging.
 - T A directory with the 'T' attribute will be deemed to be the top of directory hierarchies for the purposes of the Orlov block allocator. This is a hint to the block allocator used by ext3 and ext4 that the subdirectories under this directory are not related, and thus should be spread apart for allocation purposes. For example it is a very good idea to set the 'T' attribute on the /home directory, so that /home/john and /home/mary are placed into separate block groups. For directories where this attribute is not set, the Orlov block allocator will try to group subdirectories closer together where possible.
 - u When a file with the 'u' attribute set is deleted, its contents are saved. This allows the user to ask for its undeletion. Note: please make sure to read the bugs and limitations section at the end of this document.
 - x A file with the 'x' requests the use of direct access (dax) mode, if the kernel supports DAX. This can be overridden by the 'dax=never' mount option. For more information see the kernel documentation for dax: <https://www.kernel.org/doc/html/latest/filesystems/dax.html>.
- If the attribute is set on an existing directory, it will be inherited by all files and subdirectories that are subsequently created in the directory. If an existing directory has contained some files and subdirectories, modifying the attribute on the parent directory doesn't change the attributes on these files and subdirectories.
- V A file with the 'V' attribute set has fs-verity enabled. It cannot be written to, and the file system will automatically verify all data read from it against a cryptographic hash that covers the entire file's contents, e.g. via a Merkle tree. This makes it possible to efficiently authenticate the file. This attribute may not be set or cleared using `chattr(1)`, although it can be displayed by `lsattr(1)`.

chattr was written by Remy Card <Remy.Card@linux.org>. It is currently being maintained by Theodore Ts'o <tytso@alum.mit.edu>.

BUGS AND LIMITATIONS

The 'c', 's', and 'u' attributes are not honored by the ext2, ext3, and ext4 file systems as implemented in the current mainline Linux kernels. Setting 'a' and 'i' attributes will not affect the ability to write to already existing file descriptors.

The 'j' option is only useful for ext3 and ext4 file systems.

The 'D' option is only useful on Linux kernel 2.5.19 and later.

AVAILABILITY

chattr is part of the e2fsprogs package and is available from <http://e2fsprogs.sourceforge.net>.

SEE ALSO

lsattr(1), btrfs(5), ext4(5), mkfs.f2fs(8), xfs(5).

E2fsprogs version 1.47.2

January 2025

CHATTR(1)