



Rocky Enterprise Linux 9.2 Manual Pages on command 'lz4c.1'

C:\>man lz4c.1

LZ4(1) User Commands LZ4(1)

NAME

lz4 - lz4, unlz4, lz4cat - Compress or decompress .lz4 files

SYNOPSIS

lz4 [OPTIONS] [-|INPUT-FILE] OUTPUT-FILE

unlz4 is equivalent to lz4 -d

lz4cat is equivalent to lz4 -dcfm

When writing scripts that need to decompress files, it is recommended to always use the name lz4 with appropriate arguments (lz4 -d or lz4 -dc) instead of the names unlz4 and lz4cat.

DESCRIPTION

lz4 is an extremely fast lossless compression algorithm, based on byte-aligned LZ77 family of compression scheme. lz4 offers compression speeds of 400 MB/s per core, linearly scalable with multi-core CPUs. It features an extremely fast decoder, with speed in multiple GB/s per core, typically reaching RAM speed limit on multi-core systems. The native file format is the .lz4 format.

Difference between lz4 and gzip

lz4 supports a command line syntax similar but not identical to gzip(1). Differ?

ences are :

? lz4 compresses a single file by default (see -m for multiple files)

? lz4 file1 file2 means : compress file1 into file2

? lz4 file.lz4 will default to decompression (use -z to force compression)

- ? lz4 preserves original files
- ? lz4 shows real-time notification statistics during compression or decompression of a single file (use -q to silence them)
- ? When no destination is specified, result is sent on implicit output, which depends on stdout status. When stdout is Not the console, it becomes the implicit output. Otherwise, if stdout is the console, the implicit output is file? name.lz4.
- ? It is considered bad practice to rely on implicit output in scripts. because the script?s environment may change. Always use explicit output in scripts. -c ensures that output will be stdout. Conversely, providing a destination name, or using -m ensures that the output will be either the specified name, or file? name.lz4 respectively.

Default behaviors can be modified by opt-in commands, detailed below.

- ? lz4 -m makes it possible to provide multiple input filenames, which will be compressed into files using suffix .lz4. Progress notifications become disabled by default (use -v to enable them). This mode has a behavior which more closely mimics gzip command line, with the main remaining difference being that source files are preserved by default.
- ? Similarly, lz4 -m -d can decompress multiple *.lz4 files.
- ? It?s possible to opt-in to erase source files on successful compression or de? compression, using --rm command.
- ? Consequently, lz4 -m --rm behaves the same as gzip.

Concatenation of .lz4 files

It is possible to concatenate .lz4 files as is. lz4 will decompress such files as if they were a single .lz4 file. For example:

```
lz4 file1 > foo.lz4
lz4 file2 >> foo.lz4
```

Then lz4cat foo.lz4 is equivalent to cat file1 file2.

OPTIONS

Short commands concatenation

In some cases, some options can be expressed using short command -x or long command --long-word. Short commands can be concatenated together. For example, -d -c is equivalent to -dc. Long commands cannot be concatenated. They must be clearly sepa?

rated by a space.

Multiple commands

When multiple contradictory commands are issued on a same command line, only the latest one will be applied.

Operation mode

`-z --compress`

Compress. This is the default operation mode when no operation mode option is specified, no other operation mode is implied from the command name (for example, `unlz4` implies `--decompress`), nor from the input file name (for example, a file extension `.lz4` implies `--decompress` by default). `-z` can also be used to force compression of an already compressed `.lz4` file.

`-d --decompress --uncompress`

Decompress. `--decompress` is also the default operation when the input file name has an `.lz4` extension.

`-t --test`

Test the integrity of compressed `.lz4` files. The decompressed data is discarded. No files are created nor removed.

`-b#` Benchmark mode, using `#` compression level.

`--list` List information about `.lz4` files. note : current implementation is limited to single-frame `.lz4` files.

Operation modifiers

`-#` Compression level, with `#` being any value from 1 to 12. Higher values trade compression speed for compression ratio. Values above 12 are considered the same as 12. Recommended values are 1 for fast compression (default), and 9 for high compression. Speed/compression trade-off will vary depending on data to compress. Decompression speed remains fast at all settings.

`--fast[=#]`

Switch to ultra-fast compression levels. The higher the value, the faster the compression speed, at the cost of some compression ratio. If `=#` is not present, it defaults to 1. This setting overrides compression level if one was set previously. Similarly, if a compression level is set after `--fast`, it overrides it.

`--best` Set highest compression level. Same as `-12`.

`--favor-decSpeed`

Generate compressed data optimized for decompression speed. Compressed data will be larger as a consequence (typically by ~0.5%), while decompression speed will be improved by 5-20%, depending on use cases. This option only works in combination with very high compression levels (≥ 10).

`-D dictionaryName`

Compress, decompress or benchmark using dictionary `dictionaryName`. Compression and decompression must use the same dictionary to be compatible. Using a different dictionary during decompression will either abort due to decompression error, or generate a checksum error.

`-f --[no-]force`

This option has several effects:

If the target file already exists, overwrite it without prompting.

When used with `--decompress` and `lz4` cannot recognize the type of the source file, copy the source file as is to standard output. This allows `lz4cat --force` to be used like `cat (1)` for files that have not been compressed with `lz4`.

`-c --stdout --to-stdout`

Force write to standard output, even if it is the console.

`-m --multiple`

Multiple input files. Compressed file names will be appended a `.lz4` suffix.

This mode also reduces notification level. Can also be used to list multiple files. `lz4 -m` has a behavior equivalent to `gzip -k` (it preserves source files by default).

`-r` operate recursively on directories. This mode also sets `-m` (multiple input files).

`-B#` Block size [4-7](default : 7)

`-B4= 64KB ; -B5= 256KB ; -B6= 1MB ; -B7= 4MB`

`-BI` Produce independent blocks (default)

`-BD` Blocks depend on predecessors (improves compression ratio, more noticeable on small blocks)

`--[no-]frame-crc`

Select frame checksum (default:enabled)

--[no-]content-size

Header includes original size (default: not present)

Note : this option can only be activated when the original size can be determined, hence for a file. It won't work with unknown source size, such as stdin or pipe.

--[no-]sparse

Sparse mode support (default: enabled on file, disabled on stdout)

-l Use Legacy format (typically for Linux Kernel compression)

Note : -l is not compatible with -m (--multiple) nor -r

Other options

-v --verbose

Verbose mode

-q --quiet

Suppress warnings and real-time statistics; specify twice to suppress errors too

-h -H --help

Display help/long help and exit

-V --version

Display Version number and exit

-k --keep

Preserve source files (default behavior)

--rm Delete source files on successful compression or decompression

-- Treat all subsequent arguments as files

Benchmark mode

-b# Benchmark file(s), using # compression level

-e# Benchmark multiple compression levels, from b# to e# (included)

-i# Minimum evaluation time in seconds [1-9] (default : 3)

BUGS

Report bugs at: <https://github.com/lz4/lz4/issues>

AUTHOR

Yann Collet