



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

\$ man mksquashfs.1

MKSQUASHFS(1) User Commands MKSQUASHFS(1)

NAME

mksquashfs - tool to create and append to squashfs filesystems

SYNOPSIS

mksquashfs source1 source2 ... FILESYSTEM [OPTIONS] [-e list of exclude dirs/files]

DESCRIPTION

Squashfs is a highly compressed read-only filesystem for Linux. It uses either gzip/xz/lzo/lz4/zstd compression to compress both files, inodes and directories. Inodes in the system are very small and all blocks are packed to minimise data overhead. Block sizes greater than 4K are supported up to a maximum of 1Mbytes (default block size 128K).

Squashfs is intended for general read-only filesystem use, for archival use (i.e. in cases where a .tar.gz file may be used), and in constrained block device/memory systems (e.g. embedded systems) where low overhead is needed.

OPTIONS

Filesystem compression options:

-b BLOCK-SIZE

set data block to BLOCK-SIZE. Default 128 Kbytes. Optionally a suffix of K, KB, Kbytes or M, MB, Mbytes can be given to specify Kbytes or Mbytes respectively.

-comp COMP

select COMP compression. Run `-help-comp COMP` to get compressor options for COMP, or ALL for all the compressors. Compressors available: gzip (default), lzo, lz4, xz, zstd, lzma.

-noI do not compress inode table.

- nold do not compress the uid/gid table (implied by -nol).
- noD do not compress data blocks.
- noF do not compress fragment blocks.
- noX do not compress extended attributes.
- no-compression
 - do not compress any of the data or metadata. This is equivalent to specifying -nol
 - noD -noF and -noX.

Filesystem build options:

- tar read uncompressed tar file from standard in (stdin).
- no-strip
 - act like tar, and do not strip leading directories from source files.
- tarstyle
 - alternative name for -no-strip.
- cpiostyle
 - act like cpio, and read file pathnames from standard in (stdin).
- cpiostyle0
 - like -cpiostyle, but filenames are null terminated. Can be used with find -print0 action.
- no-exports
 - do not make filesystem exportable via NFS (-tar default).
- exports
 - make filesystem exportable via NFS (default).
- no-sparse
 - do not detect sparse files.
- no-tailends
 - do not pack tail ends into fragments (default).
- tailends
 - pack tail ends into fragments.
- no-fragments
 - do not use fragments.
- no-duplicates
 - do not perform duplicate checking.
- no-hardlinks

do not hardlink files, instead store duplicates.

-keep-as-directory

if one source directory is specified, create a root directory containing that directory, rather than the contents of the directory.

Filesystem time options:

-mkfs-time TIME

set filesystem creation timestamp to TIME. TIME can be "inode", which means use the latest inode timestamp, an unsigned 32-bit int indicating seconds since the epoch (1970-01-01) or a string value which is passed to the "date" command to parse. Any string value which the date command recognises can be used such as "now", "last week", or "Wed Feb 15 21:02:39 GMT 2023".

-inode-time TIME

set all file and directory timestamps to TIME. TIME can be "inode", which means use the latest inode timestamp, an unsigned 32-bit int indicating seconds since the epoch (1970-01-01) or a string value which is passed to the "date" command to parse. Any string value which the date command recognises can be used such as "now", "last week", or "Wed Feb 15 21:02:39 GMT 2025". This option sets and overrides the -root-time option.

-root-time TIME

set root directory time to TIME. TIME can be "inode", which means use the latest inode timestamp, an unsigned 32-bit int indicating seconds since the epoch (1970-01-01) or a string value which is passed to the "date" command to parse. Any string value which the date command recognises can be used such as "now", "last week", or "Wed Feb 15 21:02:39 GMT 2025".

-repro build a reproducible filesystem image. This is equivalent to -mkfs-time inode, which achieves reproducibility by setting the filesystem build time to the latest inode timestamp. Obviously the image won't be reproducible if the timestamps or content changes.

-repro-time TIME

build a reproducible filesystem image. This is equivalent to specifying -mkfs-time TIME and -inode-time TIME, which achieves reproducibility by setting all timestamps to TIME. This option can be used in cases where timestamps may change, and where -repro cannot be used for this reason.

Filesystem permissions options:

-all-root

make all files and directories owned by root.

-root-mode MODE

set root directory permissions to MODE. MODE can be symbolic or octal (see section Symbolic mode specification). Default root mode is ugo=rwx or 0777 octal.

-root-uid USER

set root directory owner to specified USER, USER can be either an integer uid or user name.

-root-gid GROUP

set root directory group to specified GROUP, GROUP can be either an integer gid or group name.

-force-file-mode MODE

set all file (non-directory) permissions to MODE. MODE can be symbolic or octal (see section Symbolic mode specification).

-force-dir-mode MODE

set all directory permissions to MODE. MODE can be symbolic or octal (see section Symbolic mode specification).

-force-uid USER

set all file and directory uids to specified USER, USER can be either an integer uid or user name.

-force-gid GROUP

set all file and directory gids to specified GROUP, GROUP can be either an integer gid or group name.

-uid-gid-offset VALUE

offset all uid and gids by specified VALUE. VALUE should be a positive integer.

Filesystem pseudo options:

-p PSEUDO-DEFINITION

add pseudo file definition. The definition should be quoted. See section "Pseudo file definition format" for format details.

-pd "d mode uid gid"

specify a default pseudo directory which will be used in pseudo definitions if a directory in the pathname does not exist. This also allows pseudo definitions to be

specified without specifying all the directories in the pathname. The definition should be quoted.

-pd "D time mode uid gid"

specify a default pseudo directory which will be used in pseudo definitions if a directory in the pathname does not exist. The D type also allows a timestamp to be specified in addition to mode, uid and gid.

-pf PSEUDO-FILE

add list of pseudo file definitions from PSEUDO-FILE, use - for stdin. Pseudo file definitions should not be quoted.

-pseudo-override

make pseudo file uids and gids override -all-root, -force-uid and -force-gid options.

Filesystem filter options:

-sort SORT-FILE

sort files according to priorities in SORT-FILE. One file or dir with priority per line. Priority -32768 to 32767, default priority 0.

-ef EXCLUDE-FILE

list of exclude dirs/files. One per line.

-wildcards

allow extended shell wildcards (globbing) to be used in exclude dirs/files.

-regex allow POSIX regular expressions to be used in exclude dirs/files.

-max-depth LEVELS

descend at most LEVELS of directories when scanning filesystem.

-one-file-system

do not cross filesystem boundaries. If a directory crosses the boundary, create an empty directory for each mount point. If a file crosses the boundary ignore it.

-one-file-system-x

do not cross filesystem boundaries. Like -one-file-system option except directories are also ignored if they cross the boundary.

Filesystem extended attribute (xattrs) options:

-no-xattrs

do not store extended attributes.

-xattrs

store extended attributes (default).

`-xattrs-exclude REGEX`

exclude any xattr names matching REGEX. REGEX is a POSIX regular expression, e.g.

`-xattrs-exclude '^user.'` excludes xattrs from the user namespace.

`-xattrs-include REGEX`

include any xattr names matching REGEX. REGEX is a POSIX regular expression, e.g.

`-xattrs-include '^user.'` includes xattrs from the user namespace.

`-xattrs-add NAME=VAL`

add the xattr NAME with VAL to files. If an user xattr it will be added to regular files and directories (see man 7 xattr). Otherwise it will be added to all files.

VAL by default will be treated as binary (i.e. an uninterpreted byte sequence), but

it can be prefixed with 0s, where it will be treated as base64 encoded, or prefixed

with 0x, where val will be treated as hexadecimal. Additionally it can be prefixed

with 0t where this encoding is similar to binary encoding, except backslashes are

specially treated, and a backslash followed by 3 octal digits can be used to encode

any ASCII character, which obviously can be used to encode control codes. The option

can be repeated multiple times to add multiple xattrs.

Mksquashfs runtime options:

`-version`

print version, licence and copyright message.

`-exit-on-error`

treat normally ignored errors as fatal.

`-quiet` no verbose output.

`-info` print files written to filesystem to stdout. This automatically disables the

progress bar. See `-info-file` to output to file without disabling the progress bar.

`-info-file FILE`

print files written to filesystem to file FILE. This does not disable the progress bar.

`-no-progress`

do not display the progress bar.

`-progress`

display progress bar when using the `-info` option.

`-percentage`

display a percentage rather than the full progress bar. Can be used with dialog

--gauge etc.

-throttle PERCENTAGE

throttle the I/O input rate by the given percentage. This can be used to reduce the I/O and CPU consumption of Mksquashfs.

-limit PERCENTAGE

limit the I/O input rate to the given percentage. This can be used to reduce the I/O and CPU consumption of Mksquashfs (alternative to -throttle).

-processors NUMBER

use NUMBER processors. By default will use number of processors available.

-mem SIZE

use SIZE physical memory for caches. Use K, M or G to specify Kbytes, Mbytes or Gbytes respectively.

-mem-percent PERCENT

use PERCENT physical memory for caches. Default 25%.

-mem-default

print default memory usage in Mbytes.

-single-reader

use a single thread to read files. This reads files sequentially from the source(s).

-small-readers N

use N threads to read small files (files less than a block size) in parallel from the source(s) (default 4).

-block-readers N

use N threads to read block files (files a block or larger in size) in parallel from the source(s) (default 4).

-overcommit PERCENT

allow PERCENT more threads to run in parallel than available processors. Doing this may increase CPU utilisation. Default is 0%, because normally overcommitting reduces performance due to trashing. The percentage value is at the granularity of the number of processors, e.g. 4 processors have a percentage granularity of 25%, and 20 processors have a percentage granularity of 5%.

Filesystem append options:

-noappend

do not append to existing filesystem.

-root-becomes NAME

when appending source files/directories, make the original root become a subdirectory in the new root called NAME, rather than adding the new source items to the original root.

-no-recovery

do not generate a recovery file.

-recovery-path NAME

use NAME as the directory to store the recovery file.

-recover NAME

recover filesystem data using recovery file NAME.

Filesystem actions options:

-action ACTION@EXPRESSION

evaluate EXPRESSION on every file and directory, and execute ACTION if it returns TRUE.

-log-action ACTION@EXPRESSION

as -action, but log expression evaluation results and actions performed.

-true-action ACTION@EXPRESSION

as -action, but only log expressions which return TRUE.

-false-action ACTION@EXPRESSION

as -action, but only log expressions which return FALSE.

-action-file FILE

as action, but read actions from FILE.

-log-action-file FILE

as -log-action, but read actions from FILE.

-true-action-file FILE

as -true-action, but read actions from FILE.

-false-action-file FILE

as -false-action, but read actions from FILE.

Tar file only options:

-default-mode MODE

tar files often do not store permissions for intermediate directories. This option sets the default directory permissions to MODE. MODE can be symbolic or octal (see section Symbolic mode specification). Default mode is u=rwx,go=rx or 0755 octal.

This also sets the root directory mode.

-default-uid VALUE

tar files often do not store uids for intermediate directories. This option sets the default directory owner to VALUE, rather than the user running Mksquashfs. VALUE can be either an integer uid or user name. This also sets the root directory uid.

-default-gid VALUE

tar files often do not store gids for intermediate directories. This option sets the default directory group to VALUE, rather than the group of the user running Mksquashfs. VALUE can be either an integer uid or group name. This also sets the root directory gid.

-ignore-zeros

allow tar files to be concatenated together and fed to Mksquashfs. Normally a tarfile has two consecutive 512 byte blocks filled with zeros which means EOF and Mksquashfs will stop reading after the first tar file on encountering them. This option makes Mksquashfs ignore the zero filled blocks.

-numeric-owner

Use the numeric uid and gid in the tarfile rather than the user name or group name. This is useful where the user name or group name maps to a different uid or gid on the system where Mksquashfs is being run, and this remapping behaviour is not wanted.

Expert options (these may make the filesystem unmountable):

-stream

output the filesystem to STDOUT rather than to a file. This allows the output to be piped to another program or elsewhere with ssh. The resultant streamed Squashfs filesystem will not be a normal Squashfs filesystem because the super block will be at the end of the filesystem. But it can be fixed up afterwards with the -fix option and Unsquashfs 4.7.3 and later can read streamed unfixed up filesystems directly. Use - for the output FILESYSTEM on the command line. Duplicate detection is disabled using the -stream option.

-fix FILESYSTEM

fix FILESYSTEM generated using the -stream option.

-nopad do not pad filesystem to a multiple of 4K.

-offset OFFSET

skip OFFSET bytes at the beginning of FILESYSTEM. Optionally a suffix of K, M or G

can be given to specify Kbytes, Mbytes or Gbytes respectively. Default 0 bytes.

-o OFFSET

synonym for -offset.

Help options:

-help print help summary information to pager (or stdout if not a terminal).

-help-option REGEX

print the help information for options matching REGEX to pager (or stdout if not a terminal).

-help-section SECTION

print the help information for section SECTION to pager (or stdout if not a terminal). If SECTION does not exactly match a section name, it is treated as a regular expression, and all section names that match are displayed. Use "list" as section name to get a list of sections and their names.

-help-comp COMP

print compressor options for compressor COMP. Use "list" to get a list of available compressors, and "all" to get the compressor options for all the compressors.

-help-all

print help information for all Mksquashfs options and sections to pager (or stdout if not a terminal).

-Xhelp print compressor options for selected compressor.

-h shorthand alternative to -help.

-ho REGEX

shorthand alternative to -help-option.

-hs SECTION

shorthand alternative to -help-section.

-ha shorthand alternative to -help-all.

-no-pager

do not use a pager to output help information.

-cols WIDTH

use WIDTH columns to output help information. Useful if output is not to a terminal.

Miscellaneous options:

-fstime TIME

alternative name for -mkfs-time.

-always-use-fragments

alternative name for -tailends.

-root-owned

alternative name for -all-root.

-noInodeCompression

alternative name for -noI.

-noIdTableCompression

alternative name for -noId.

-noDataCompression

alternative name for -noD.

-noFragmentCompression

alternative name for -noF.

-noXattrCompression

alternative name for -noX.

-pseudo-dir

alternative name for -pd.

PSEUDO FILE DEFINITION FORMAT

-p "filename d mode uid gid"

create a directory.

-p "filename m mode uid gid"

modify filename.

-p "filename b mode uid gid major minor"

create a block device.

-p "filename c mode uid gid major minor"

create a character device.

-p "filename f mode uid gid command"

create file from stdout of command.

-p "filename s mode uid gid symlink"

create a symbolic link.

-p "filename i mode uid gid [s|f]"

create a socket (s) or FIFO (f).

-p "filename x name=val"

create an extended attribute.

-p "filename h linkname"

create a hard-link to linkname, follows symlinks.

-p "filename l linkname"

create a hard-link to linkname.

-p "filename L pseudo_filename"

same, but link to pseudo file.

-p "filename D time mode uid gid"

create a directory with timestamp time.

-p "filename M time mode uid gid"

modify a file with timestamp time.

-p "filename B time mode uid gid major minor"

create block device with timestamp time.

-p "filename C time mode uid gid major minor"

create char device with timestamp time.

-p "filename F time mode uid gid command"

create file with timestamp time.

-p "filename S time mode uid gid symlink"

create symlink with timestamp time.

-p "filename l time mode uid gid [s|f]"

create socket/fifo with timestamp time.

SYMBOLIC MODE SPECIFICATION

The symbolic mode is of the format [ugoa]*[[+|=]PERMS]+. PERMS = [rwxXst]+ or [ugo], and the sequence can be repeated separated with commas.

A combination of the letters ugoa specify which permission bits will be affected, u means user, g means group, o means other, and a means all or ugo.

The next letter is +, - or =. The letter + means add to the existing permission bits, - means remove the bits from the existing permission bits, and = means set the permission bits.

The permission bits (PERMS) are a combination of [rwxXst] which sets/adds/removes those bits for the specified ugoa combination, r means read, w means write and x means execute for files or search for directories. X has a special meaning, if the file is a directory it is equivalent to x or search, but if it is a non-directory, it only takes effect if execute is already set for user, group or other. The s flag sets user or group ID on execution, and

the `t` flag on a directory sets restricted deletion, or historically made the file sticky if a non-directory.

The permission bits can also be `u`, `g` or `o`, which takes the permission bits from the user, group or other of the file respectively.

EXIT STATUS

- 0 Mksquashfs successfully generated a filesystem.
- 1 Fatal errors occurred, Mksquashfs aborted and did not generate a filesystem (or up? date if appending).

COMPRESSORS AVAILABLE AND COMPRESSOR SPECIFIC OPTIONS

gzip (default):

`-Xcompression-level COMPRESSION-LEVEL`

COMPRESSION-LEVEL should be 1 .. 9 (default 9).

`-Xwindow-size WINDOW-SIZE`

WINDOW-SIZE should be 8 .. 15 (default 15).

`-Xstrategy strategy1,strategy2,...,strategyN`

Compress using strategy1,strategy2,...,strategyN in turn and choose the best compression. Available strategies: default, filtered, huffman_only, run_length_encoded and fixed.

lzo:

`-Xalgorithm ALGORITHM`

Where ALGORITHM is one of: lzo1x_1, lzo1x_1_11, lzo1x_1_12, lzo1x_1_15, lzo1x_999 (default).

`-Xcompression-level COMPRESSION-LEVEL`

COMPRESSION-LEVEL should be 1 .. 9 (default 8). Only applies to lzo1x_999 algorithm.

lz4:

`-Xhc` Compress using LZ4 High Compression.

xz:

`-Xbcj filter1,filter2,...,filterN`

Compress using filter1,filter2,...,filterN in turn (in addition to no filter), and choose the best compression. Available filters: x86, arm, armthumb, arm64, powerpc, sparc, ia64, riscv.

`-Xdict-size DICT-SIZE`

Use DICT-SIZE as the XZ dictionary size. The dictionary size can be specified as a

percentage of the block size, or as an absolute value. The dictionary size must be less than or equal to the block size and 8192 bytes or larger. It must also be storable in the xz header as either 2^n or as $2^n + 2^{(n+1)}$. Example dict-sizes are 75%, 50%, 37.5%, 25%, or 32K, 16K, 8K etc.

zstd:

-Xcompression-level COMPRESSION-LEVEL

COMPRESSION-LEVEL should be -131072 .. -1 or 1 .. 22 (default 15). Negative compression levels correspond to the zstd --fast option.

lzma:

(no options) (deprecated - no kernel support)

ENVIRONMENT

SQFS_CMDLINE

If set, this is used as the directory to write the file sqfs_cmdline which contains the command line arguments given to Mksquashfs. Each command line argument is wrapped in quotes to ensure there is no ambiguity when arguments contain spaces. If the file already exists then the command line is appended to the file.

SOURCE_DATE_EPOCH

If set, this is used as the filesystem creation timestamp. Also any file timestamps which are after SOURCE_DATE_EPOCH will be clamped to SOURCE_DATE_EPOCH. See <https://reproducible-builds.org/docs/source-date-epoch/> for more information.

PAGER If set, this is used as the name of the program used to display the help text. The value can be a simple command or a pathname. The default is /usr/bin/pager.

EXAMPLES

mksquashfs DIRECTORY IMAGE.SQFS

Create a Squashfs filesystem from the contents of DIRECTORY, writing the output to IMAGE.SQFS. Mksquashfs will use the default compressor (normally gzip), and a block size of 128 Kbytes.

mksquashfs DIRECTORY FILE1 FILE2 IMAGE.SQFS

Create a Squashfs filesystem containing DIRECTORY and FILE1 and FILE2. If multiple sources are specified on the command line they will be combined into a single directory.

mksquashfs DIRECTORY IMAGE.SQFS -b 1M -comp zstd

Use a block size of 1 Mbyte and Zstandard compression to create the filesystem.

`mksquashfs DIRECTORY IMAGE.SQFS -all-time now`

Set all file and directory timestamps to "now" (current time) in the filesystem.

`mksquashfs DIRECTORY IMAGE.SQFS -force-uid phillip -force-gid phillip`

Make all files and directories in the filesystem owned by "phillip".

`mksquashfs DIRECTORY IMAGE.SQFS -force-file-mode ugo+r,go-w`

Make all files readable by everyone, but only writable by user, all other permission bits are unaffected.

`mksquashfs DIRECTORY IMAGE.SQFS -e file1 file2`

Exclude file1 and file2 from DIRECTORY when creating filesystem. No wildcard matching of files.

`mksquashfs DIRECTORY IMAGE.SQFS -wildcards -e "*.gz"`

Exclude anything in DIRECTORY which matches the wildcard pattern "*.gz".

`mksquashfs DIRECTORY IMAGE.SQFS -wildcards -e "... *.gz"`

Exclude files which match the wildcard pattern "*.gz" anywhere within DIRECTORY and its sub-directories. The initial "..." indicates the wildcard pattern is "non-anchored" and will match anywhere.

Note: when passing wildcarded names to Mksquashfs, they should be quoted (as in the above examples), to ensure that they are not processed by the shell.

Using pseudo file definitions

`mksquashfs DIRECTORY IMAGE.SQFS -p "build_dir d 0644 0 0"`

Create a directory called "build_dir" in the output filesystem.

`mksquashfs DIRECTORY IMAGE.SQFS -p "version.txt l /tmp/build/version"`

Create a reference called "version.txt" to a file outside DIRECTORY, which acts as if the file "/tmp/build/version" was copied or hard-linked into DIRECTORY before calling Mksquashfs.

`mksquashfs DIRECTORY IMAGE.SQFS -p "date.txt f 0644 0 0 date"`

Create a file called "date.txt" which holds the output (stdout) from running the "date" command.

`mksquashfs DIRECTORY IMAGE.SQFS -p "\"hello world\" f 0644 0 0 date"`

As above, but, showing that filenames can have spaces, if they are quoted. The quotes need to be backslashed to protect them from the shell.

`mksquashfs - IMAGE.SQFS -p "input f 0644 root root dd if=/dev/sda1 bs=1024" -p "/" d 0644 0`

0"

Create a file containing the contents of partition /dev/sda1". Ordinarily Mksquashfs given a device, fifo, or named socket will place that special file within the Squashfs filesystem, the above allows input from these special files to be captured and placed in the Squashfs filesystem. Note there are no other sources than the pseudo file, and so the command line source is "-". If there are no other sources than pseudo files, the root (/) directory must be defined too, as seen in this example.

```
unsquashfs -pf - IMAGE.SQFS | mksquashfs - NEW.SQFS -pf -
```

Transcode IMAGE.SQFS to NEW.SQFS by piping the pseudo file output from Unsquashfs to Mksquashfs using stdout and stdin. This can convert from earlier Squashfs filesystems or change compression algorithm, block size etc. without needing to unpack into an intermediate directory or file.

Note: pseudo file definitions should be quoted (as in the above examples), to ensure that they are passed to Mksquashfs as a single argument, and to ensure that they are not processed by the shell.

Using extended attribute options

```
mksquashfs DIRECTORY IMAGE.SQFS -no-xattrs
```

Do not store any extended attributes in the Squashfs filesystem. Any extended attributes in the source files will be ignored.

```
mksquashfs DIRECTORY IMAGE.SQFS -xattrs-include "^user."
```

Filter the extended attributes in the source files, and only store extended attributes in the user namespace in the Squashfs filesystem.

```
mksquashfs DIRECTORY IMAGE.SQFS -xattrs-exclude "^user."
```

Filter the extended attributes in the source files, and don't store any extended attributes in the user namespace in the Squashfs filesystem.

```
mksquashfs DIRECTORY IMAGE.SQFS -xattrs-add "user.comment=hello world"
```

Add the extended attribute called "user.comment" with the content "hello world" to all files and directories in the Squashfs filesystem.

```
mksquashfs DIRECTORY IMAGE.SQFS -xattrs-add "user.comment=0thello world\012"
```

Add the extended attribute called "user.comment" to all files and directories, but in this case the contents of the extended attribute will be "hello world" with a trailing newline character (012 octal).

```
mksquashfs DIRECTORY IMAGE.SQFS -xattrs-add "user.comment=0saGVsbG8gd29ybGQ="
```

Add the extended attribute called "user.comment" to all files and directories, where the value is given in base64 encoding, representing "hello world".

```
mksquashfs DIRECTORY IMAGE.SQFS -action "-xattrs-include(^user.) @ type(f)"
```

Filter the extended attributes but only in regular files (type f), and only store extended attributes in the user namespace.

```
mksquashfs DIRECTORY IMAGE.SQFS -p "hello_world x user.comment=0tsalve mundi\012"
```

Add the extended attribute called "user.comment" to the file called "hello_world", with the contents of the extended attribute being "salve mundi" with a trailing new line character (012 octal).

Using Actions to not compress, change attributes etc.

```
mksquashfs DIRECTORY IMAGE.SQFS -action "uncompressed @ (name(*.jpg) || name(*.mpg) ) || (name(*.img) && filesize(>=1G))"
```

Specify that any files matching the wildcards "*.jpg" and "*.mpg" should not be compressed. Additionally, it also specifies any files matching the wildcard "*.img" and are 1 Gigabyte or larger should be uncompressed too. This shows test operators can be combined with logical expressions.

```
mksquashfs DIRECTORY IMAGE.SQFS -action "chmod(o+r)@! perm(o+r)"
```

If any files within DIRECTORY are not readable by "others", then make them readable by others in the Squashfs filesystem.

```
mksquashfs DIRECTORY IMAGE.SQFS -action "uid(phillip)@! perm(o+r)"
```

As previous, match on any files which are not readable by "others", but, in this case change the owner of the file to "phillip" in the Squashfs filesystem.

```
mksquashfs DIRECTORY IMAGE.SQFS -action "prune @ type(l) && ! exists"
```

Delete any symbolic link within DIRECTORY which points outside of DIRECTORY, i.e. will be unresolvable in the Squashfs filesystem.

```
mksquashfs DIRECTORY IMAGE.SQFS -action "exclude @ depth(3)"
```

Create a Squashfs filesystem containing the two top most levels (contents of DIRECTORY and immediate sub-directories), and exclude anything at level 3 or below.

```
mksquashfs DIRECTORY IMAGE.SQFS -action "-xattrs-include(^user.) @ type(f)"
```

Filter the extended attributes but only in regular files (type f), and only store extended attributes in the user namespace.

Note: actions should be quoted (as in the above examples), to ensure that they are passed to Mksquashfs as a single argument, and to ensure that they are not processed by the shell.

AUTHOR

Written by Phillip Lougher <phillip@squashfs.org.uk>

COPYRIGHT

Copyright ? 2026 Phillip Lougher <phillip@squashfs.org.uk>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

See the GNU General Public License for more details.

SEE ALSO

unsquashfs(1), sqfstar(1), sqfscat(1)

The README for the Squashfs-tools 4.7.5 release, describing the new features can be read here <https://github.com/plougher/squashfs-tools/blob/master/Documentation/4.7.5/README.md>

The MKSQUASHFS usage guide can be read here <https://github.com/plougher/squashfs-tools/blob/master/Documentation/4.7.5/USAGE-MK?SQUASHFS.md>

The CHANGELOG for Squashfs tools can be read here <https://github.com/plougher/squashfs-tools/blob/master/CHANGES.md>

The ACTIONS-README for the actions subsystem can be read here <https://github.com/plougher/squashfs-tools/blob/master/Documentation/4.7.5/ACTIONS-README.md>

mksquashfs version 4.7.5

March 2026

MKSQUASHFS(1)