NAME

fsck.fat – check and repair MS-DOS FAT filesystems

SYNOPSIS

fsck.fat [*OPTIONS*] *DEVICE*

DESCRIPTION

fsck.fat verifies the consistency of MS-DOS filesystems and optionally tries to repair them.

The following filesystem problems can be corrected (in this order):

- FAT contains invalid cluster numbers. Cluster is changed to EOF.

- File's cluster chain contains a loop. The loop is broken.

- Bad clusters (read errors). The clusters are marked bad and they are removed from files owning them. This check is optional.

- Directories with a large number of bad entries (probably corrupt). The directory can be deleted.

- Files . and .. are non-directories. They can be deleted or renamed.

- Directories . and .. in root directory. They are deleted.

- Bad filenames. They can be renamed.

- Duplicate directory entries. They can be deleted or renamed.

- Directories with non-zero size field. Size is set to zero.

- Directory . does not point to parent directory. The start pointer is adjusted.

- Directory .. does not point to parent of parent directory. The start pointer is adjusted.

- . and .. are not the two first entries in a non-root directory. The entries are created, moving occupied slots if necessary.

- Start cluster number of a file is invalid. The file is truncated.

- File contains bad or free clusters. The file is truncated.

- File's cluster chain is longer than indicated by the size fields. The file is truncated.

- Two or more files share the same cluster(s). All but one of the files are truncated. If the file being truncated is a directory file that has already been read, the filesystem check is restarted after truncation.

- File's cluster chain is shorter than indicated by the size fields. The file is truncated.

- Volume label in root directory or label in boot sector is invalid. Invalid labels are removed.

- Volume label in root directory and label in boot sector are different. Volume label from root directory is copied to boot sector.

- Clusters are marked as used but are not owned by a file. They are marked as free.

Additionally, the following problems are detected, but not repaired:

- Invalid parameters in boot sector

When **fsck.fat** checks a filesystem, it accumulates all changes in memory and performs them only after all checks are complete. This can be disabled with the **−w** option.

Two different variants of the FAT filesystem are supported. Standard is the FAT12, FAT16 and FAT32 filesystems as defined by Microsoft and widely used on hard disks and removable media like USB sticks and SD cards. The other is the legacy Atari variant used on Atari ST.

There are some minor differences in Atari format: Some boot sector fields are interpreted slightly different, and the special FAT entries for end-of-file and bad cluster can be different. Under MS-DOS 0xfff8 is used for EOF and Atari employs 0xffff by default, but both systems recognize all values from 0xfff8–0xffff as end-of-file. MS-DOS uses only 0xfff7 for bad clusters, where on Atari values 0xfff0–0xfff7 are for this

purpose (but the standard value is still 0xfff7).

OPTIONS

−a   Automatically repair the filesystem. No user intervention is necessary. Whenever there is more than one method to solve a problem, the least destructive approach is used.

−A   Select using the Atari variation of the FAT filesystem if that isn't active already, otherwise select standard FAT filesystem. This is selected by default if **mkfs.fat** is run on 68k Atari Linux.

-b   Make read-only boot sector check.

-c *PAGE*
    Use DOS codepage *PAGE* to decode short file names. By default codepage 850 is used.

−d *PATH*
    Delete the specified file. If more than one file with that name exist, the first one is deleted. This option can be given more than once.

−f   Salvage unused cluster chains to files. By default, unused clusters are added to the free disk space except in auto mode (**−a**).

−F *NUM*
    Specify FAT table *NUM* for filesystem access. By default value *0* is assumed and then the first uncorrupted FAT table is chosen. Uncorrupted means that FAT table has valid first cluster. If default value *0* is used and all FAT tables are corrupted then **fsck.fat** gives up and does not try to repair FAT filesystem. If non-zero *NUM* value is specified then **fsck.fat** uses FAT table *NUM* for repairing FAT filesystem. If FAT table *NUM* has corrupted first cluster then **fsck.fat** will repair it. In any case, if FAT filesystem has more FAT tables then repaired content of chosen FAT table is copied to other FAT tables. To repair corrupted first cluster it is required to call **fsck.fat** with non-zero *NUM* value.

−l   List path names of files being processed.

−n   No-operation mode: non-interactively check for errors, but don't write anything to the filesystem.

−p   Same as **−a**, for compatibility with other *fsck.

−r   Interactively repair the filesystem. The user is asked for advice whenever there is more than one approach to fix an inconsistency. This is the default mode and the option is only retained for backwards compatibility.

−S   Consider short (8.3) file names with spaces in the middle to be invalid, like previous versions of this program did. While such file names are not forbidden by the FAT specification, and were never treated as errors by Microsoft file system checking tools, many DOS programs are unable to handle files with such names. Using this option can make them accessible to these programs.

    Short file names which *start* with a space are considered invalid regardless of this option's setting.

    Previous versions of this program exceptionally treated *EA DATA. SF* and *WP ROOT. SF* as valid short names; using this option does not preserve that exception.

−t   Mark unreadable clusters as bad.

−u *PATH*
    Try to undelete the specified file. **fsck.fat** tries to allocate a chain of contiguous unallocated clusters beginning with the start cluster of the undeleted file. This option can be given more than once.

−U   Consider lowercase volume and boot label as invalid and allow only uppercase characters. Such labels are forbidden by the FAT specification, but they are widely used by Linux tools. Moreover MS-DOS and Windows systems do not have problems to read them. Therefore volume and boot labels with lowercase characters are by default permitted.

−v   Verbose mode. Generates slightly more output.

−V   Perform a verification pass. The filesystem check is repeated after the first run. The second pass should never report any fixable errors. It may take considerably longer than the first pass, because the first pass may have generated long list of modifications that have to be scanned for each disk read.

−−variant *TYPE*
> Create a filesystem of variant *TYPE*. Acceptable values are *standard* and *atari* (in any combination of upper/lower case). See above under DESCRIPTION for the differences.

−w Write changes to disk immediately.

−y Same as **−a** (automatically repair filesystem) for compatibility with other fsck tools.

−−help
> Display help message describing usage and options then exit.

EXIT STATUS

0   No recoverable errors have been detected.

1   Recoverable errors have been detected or **fsck.fat** has discovered an internal inconsistency.

2   Usage error. **fsck.fat** did not access the filesystem.

FILES

fsck0000.rec, *fsck0001.rec*, ...
> When recovering from a corrupted filesystem, **fsck.fat** dumps recovered data into files named *fsckNNNN.rec* in the top level directory of the filesystem.

BUGS

- Does not remove entirely empty directories.

- Should give more diagnostic messages.

- Undeleting files should use a more sophisticated algorithm.

SEE ALSO

fatlabel(8), **mkfs.fat**(8)

HOMEPAGE

The home for the **dosfstools** project is its GitHub project page ⟨https://github.com/dosfstools/dosfstools⟩.

AUTHORS

dosfstools were written by Werner Almesberger ⟨werner.almesberger@lrc.di.epfl.ch⟩, Roman Hodek ⟨Roman.Hodek@informatik.uni-erlangen.de⟩, and others. Current maintainers are Andreas Bombe ⟨aeb@debian.org⟩ and Pali Rohár ⟨pali.rohar@gmail.com⟩.