



**Full credit is given to the above companies including the Operating System (OS) that this PDF file was generated!**

### ***Rocky Enterprise Linux 9.2 Manual Pages on command 'dosfstool.8'***

#### ***\$ man dosfstool.8***

FATLABEL(8)                      System Manager's Manual                      FATLABEL(8)

#### NAME

fatlabel - set or get MS-DOS filesystem label or volume ID

#### SYNOPSIS

fatlabel [OPTIONS] DEVICE [NEW]

#### DESCRIPTION

fatlabel will display or change the volume label or volume ID on the MS-DOS filesystem located on DEVICE. By default it works in label mode. It can be switched to volume ID mode with the option -i or --volume-id.

If NEW is omitted, then the existing label or volume ID is written to the standard output.

A label can't be longer than 11 bytes and should be in all upper case for best compatibility.

An empty string or a label consisting only of white space is not allowed. A volume ID must be given as a hexadecimal number (no leading "0x" or similar) and must fit into 32 bits.

#### OPTIONS

-i, --volume-id

Switch to volume ID mode.

-r, --reset

Remove label in label mode or generate new ID in volume ID mode.

-c PAGE, --codepage=PAGE

Use DOS codepage PAGE to encode/decode label. By default codepage 850 is used.

-h, --help

Display a help message and terminate.

-V, --version

Show version number and terminate.

## COMPATIBILITY and BUGS

For historic reasons FAT label is stored in two different locations: in the boot sector and as a special volume label entry in the root directory. MS-DOS 5.00, MS-DOS 6.22, MS-DOS 7.10, Windows 98, Windows XP and also Windows 10 read FAT label only from the root directory. Absence of the volume label in the root directory is interpreted as empty or none label, even if boot sector contains some valid label.

When Windows XP or Windows 10 system changes a FAT label it stores it only in the root directory ? letting boot sector unchanged. Which leads to problems when a label is removed on Windows. Old label is still stored in the boot sector but is removed from the root directory.

dosfslabel prior to the version 3.0.7 operated only with FAT labels stored in the boot sector, completely ignoring a volume label in the root directory.

dosfslabel in versions 3.0.7-3.0.15 reads FAT labels from the root directory and in case of absence, it fallbacks to a label stored in the boot sector. Change operation resulted in updating a label in the boot sector and sometimes also in the root directory due to the bug. That bug was fixed in dosfslabel version 3.0.16 and since this version dosfslabel updates label in both location.

Since version 4.2, fatlabel reads a FAT label only from the root directory (like MS-DOS and Windows systems), but changes a FAT label in both locations. In version 4.2 was fixed handling of empty labels and labels which starts with a byte 0xE5. Also in this version was added support for non-ASCII labels according to the specified DOS codepage and were added checks if a new label is valid.

It is strongly suggested to not use dosfslabel prior to version 3.0.16.

## DOS CODEPAGES

MS-DOS and Windows systems use DOS (OEM) codepage for encoding and decoding FAT label. In Windows systems DOS codepage is global for all running applications and cannot be configured explicitly. It is set implicitly by option Language for non-Unicode programs available in Regional and Language Options via Control Panel. Default DOS codepage for fatlabel is 850. See following mapping table between DOS codepage and Language for non-Unicode programs:

Codepage Language

- 437 English (India), English (Malaysia), English (Republic of the Philippines),  
English (Singapore), English (South Africa), English (United States), English  
(Zimbabwe), Filipino, Hausa, Igbo, Inuktitut, Kinyarwanda, Kiswahili, Yoruba
- 720 Arabic, Dari, Persian, Urdu, Uyghur
- 737 Greek
- 775 Estonian, Latvian, Lithuanian
- 850 Afrikaans, Alsatian, Basque, Breton, Catalan, Corsican, Danish, Dutch, English  
(Australia), English (Belize), English (Canada), English (Caribbean), English  
(Ireland), English (Jamaica), English (New Zealand), English (Trinidad and To?  
bago), English (United Kingdom), Faroese, Finnish, French, Frisian, Galician,  
German, Greenlandic, Icelandic, Indonesian, Irish, isiXhosa, isiZulu, Italian,  
K'iche, Lower Sorbian, Luxembourgish, Malay, Mapudungun, Mohawk, Norwegian, Oc?  
citan, Portuguese, Quechua, Romansh, Sami, Scottish Gaelic, Sesotho sa Leboa,  
Setswana, Spanish, Swedish, Tamazight, Upper Sorbian, Welsh, Wolof
- 852 Albanian, Bosnian (Latin), Croatian, Czech, Hungarian, Polish, Romanian, Ser?  
bian (Latin), Slovak, Slovenian, Turkmen
- 855 Bosnian (Cyrillic), Serbian (Cyrillic)
- 857 Azeri (Latin), Turkish, Uzbek (Latin)
- 862 Hebrew
- 866 Azeri (Cyrillic), Bashkir, Belarusian, Bulgarian, Kyrgyz, Macedonian, Mongo?  
lian, Russian, Tajik, Tatar, Ukrainian, Uzbek (Cyrillic), Yakut
- 874 Thai
- 932 Japanese
- 936 Chinese (Simplified)
- 949 Korean
- 950 Chinese (Traditional)
- 1258 Vietnamese

#### SEE ALSO

fsck.fat(8), mkfs.fat(8)

#### HOME PAGE

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@irc.di.epfl.ch](mailto:werner.almesberger@irc.di.epfl.ch), Roman Hodek [Roman.Hodek@informatik.uni-erlangen.de](mailto:Roman.Hodek@informatik.uni-erlangen.de), and others. Current maintainers are Andreas Bombe [aeb@debian.org](mailto:aeb@debian.org) and Pali Rohar [pali.rohar@gmail.com](mailto:pali.rohar@gmail.com).

dosfstools 4.2

2021-01-31

FATLABEL(8)