## NAME

ext3Viewer − an ext3 filesystem low level viewer

## USAGE

ext3Viewer <filesystem> <command> [<arguments>]

## SYNOPSIS

**ext3Viewer** <fs> [ -stats | -stat | -printgd | -imap | -bmap |
-ffi | -ffb | -cat | -bcat | -ls |
-find | -iname | -backups | -acl | -jbd |
-bpoint | -tree ]

**'ext3viewer -help'** for help

## DESCRIPTION

ext3viewer is intended to explore an ext3 filesystem at a low level. ext3viewer lets you study all the ext3 structures and prints them in a more user-friendly form.

## OPTIONS

All options (except -help) have to follow the mandatory argument <filesystem>

-stats [ <block num> <block size> ]
: Informations about the superblock and the group descriptors. Given alone, -stats dumps the ext3_super_block structure of the first super block which is at the 1024th byte. If you specify the block number where a copy of the superblock is, and the block size of your filesystem, it will read an alternate superblock.(see also -backups). See an example in the section below.

-stat <inode num>
: This option shows specific informations for a given inode. (to obtain file's inode number, use the -ls option from ext3viewer, or mount the filesystem, and use the 'ls -li' command.)

-tree <inode num>
: With this option, you can have a partial view of the block's allocation for an inode. It will shown as a tree, with a few block numbers if the file uses indirect blocks. See -bpoint for more details.

-bpoint <block num>
: Use this option to know which indirect blocks are allocated to an inode. With the -stat option you can see the direct blocks (blocks from 0 to 11) and indirect blocks (block number 12), double indirect blocks (block number 13) and triple indirect blocks (block number 14). To see the real blocks used to store the file data you can use this -bpoint option. For example, you can use it on an indirect block (the 12th block), to see a list of the blocks that compose the file data; the double indirect blocks will require you to do this twice (and recursively), etc.

-printgd <group num>
: Print the fields of ext3_group_desc for one group. (groups start at number 0)

-imap <group_desc num>
: The inode bitmap is a bits list that indicate neither an inode is free or in use. This option shows the inode bitmap as an array. As the ouput is quite verbose, the use of the 'less' utility is recommended. ( ex: ext3viewer /dev/hda2 -imap 0 | less )

-bmap <group_desc num>
: Same as -imap, for the block bitmap.

-ffi
: Search for the first free inode on the filesystem through to the inode bitmap.

-ffb
: Search for the first free block on the filesystem through to the block bitmap.

-cat <inode num> [ -x ]
: Show the contents of a file designed by its inode number (using putchar() or print in hexadecimal if option -x is given). It reads recursively all the blocks allocated to the file. To see the contents of a directory, it is recommended to use the -ls option but using -cat on a directory can be useful to see that the directories' blocks are never freed.

-bcat <block num> [ -x ]
Show <block num>'s contents. It only reads one block unlike the -cat option. To know the block numbers allocated to an inode, see option -stat and -bpoint. If option -x is given, print the contents of the block in hexadecimal.

-ls <inode num>
Print the directory entries of an inode. A directory is an inode with blocks that contains directory entries ( ext3_dir_entry ). It only gives the directory entries' fields. To see more details about a given inode, use option -stat. File types are represented by the same letters that in 'ls -l'. - for an ordinary file, d for a directory, b for a block special device, c for a character special device, l for a symbolic link, p for a FIFO, s for a socket.

-find <inode_dir starting number> <regex pattern>
Find all the files that match with the pattern starting at a inode_dir directory. It will go through all directories and sub-directories recursively. The regex pattern is used by the regexec() POSIX function. To start seeking a file from the root, use inode number 2. (this might take some time, depending on the filesystem size)

-iname <inode num>
Find all the files that have the given inode. It can return more that one file due to multiple hard links (directory entries pointing the same inode). This might take some time, depending on the filesystem size.

-backups
Ext3 maintains several copies of the superblock since it is a very important structure. This option calculates where these copies are. You can print a superblock copy's contents using the -stats option.

-jbd Dump the journal superblock and descriptor blocks. When a filesystem is umounted, the journal is not flushed but just marked as empty. So you can examine the journal even if the filesystem is umounted. Just be aware that if two journal sequence number are not consecutive, it means the end of the journal for a session. Between two mount operations, ext3 writes into the previous journal (not appending). See /usr/include/linux/jbd.h

-help Short summary of the available options. NB: if you use an option without the appropriate arguments, it will print which arguments are used for that option.

## EXAMPLES

./ext3viewer -help
print the short help

./ext3viewer /dev/hda1 -ls 2
print the directory entries of the "/" directory, from the /dev/hda1 partition.

./ext3viewer /dev/hda2 -find 2 'toto.*'
search for all files that are named 'toto' followed by something in the whole the filesystem.

./ext3viewer /dev/hda2 -find 65537 '\<foo\>'
search for all files that are exactly named 'foo' in the directory which has the inode number 65537, and its sub−directories.

./ext3viewer /dev/hda2 -stats
show the superblock fields, using the first superblock

./ext3viewer /dev/hda2 -stats 163840 4096
show the superblock fields using a copy of the superblock on group number 5. The filesystem has a 4096-bytes block size and the superblock copy is at the 163840th block. This data has been retrieved with this command: ./ext3viewer /dev/hda2 -backups.

## DIAGNOSTICS

You must be root powered to open a device like /dev/hda1. If you are not a superuser, you can use an image made by yourself to study the ext3 filesystem (ex: dd if=/dev/zero of=filesystem.ext3 ...; mkfs.ext3

./filesystem.ext3).

The filesystem has to be unmounted to be studied safely with ext3viewer. Using it on a mounted filesystem may result in inconsistent results.

The following diagnostics may be issued on stderr:

− Bad magic number : the filesystem is not ext3 or the superbloc is corrupted. If you have enough details try using an alternate superblock copy with -stats.

− Permission denied : you do not have the right to open the filesystem. Remember that you have to be root powered to open /dev/hdXX or /dev/sdXX.

## BUGS

If you see any bug, please send us a mail at <ext3viewer@free.fr> that explains how you have discovered it.

## CREDITS

ext3Viewer is distributed under the GNU public license. See the file COPYING for details.

A WEB site is available at http://ext3viewer.free.fr

## THANKS

Thanks to Konstantin Verchinine and Andrei Paskevich from whom we have learned a lot. Thanks to Julien Poitrat for his original idea and his e2view project (2003).

## AUTHORS

−− Laurent Sebag <laurentsebag@free.fr>

−− Nathan Perianayagassamy <nathan.periana@yahoo.com>

## SEE ALSO

**debugfs**(8), **dumpe2fs**(8), **/usr/include/linux/ext3_fs.h /usr/include/linux/jbd.h**