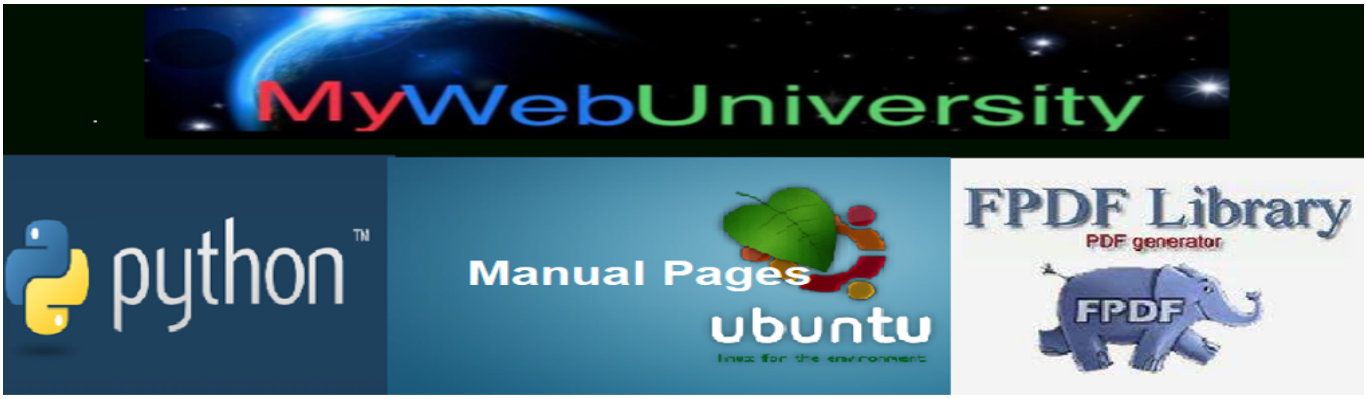

Rocky Enterprise Linux 9.2 Manual Pages on command 'e2fsck.8'

$ man e2fsck.8

E2FSCK(8) System Manager's Manual E2FSCK(8)

NAME

e2fsck - check a Linux ext2/ext3/ext4 file system

SYNOPSIS

e2fsck [-pacnyrdfkvtDFV] [-b superblock] [-B blocksize] [-l|-L bad_blocks_file] [-C fd] [-j external-journal] [-E extended_options] [-z undo_file] device

DESCRIPTION

e2fsck is used to check the ext2/ext3/ext4 family of file systems. For ext3 and ext4 file systems that use a journal, if the system has been shut down uncleanly without any errors, normally, after replaying the committed transactions in the journal, the file system should be marked as clean. Hence, for file systems that use journaling, e2fsck will nor? mally replay the journal and exit, unless its superblock indicates that further checking is required.

device is a block device (e.g., /dev/sdc1) or file containing the file system.

Note that in general it is not safe to run e2fsck on mounted file systems. The only ex? ception is if the -n option is specified, and -c, -l, or -L options are not specified.

However, even if it is safe to do so, the results printed by e2fsck are not valid if the file system is mounted. If e2fsck asks whether or not you should check a file system which is mounted, the only correct answer is ``no''. Only experts who really know what they are doing should consider answering this question in any other way.

If e2fsck is run in interactive mode (meaning that none of -y, -n, or -p are specified), the program will ask the user to fix each problem found in the file system. A response of 'y' will fix the error; 'n' will leave the error unfixed; and 'a' will fix the problem and

all subsequent problems; pressing Enter will proceed with the default response, which is printed before the question mark. Pressing Control-C terminates e2fsck immediately.

OPTIONS

-a This option does the same thing as the -p option. It is provided for backwards compatibility only; it is suggested that people use -p option whenever possible.

-b superblock

Instead of using the normal superblock, use an alternative superblock specified by superblock. This option is normally used when the primary superblock has been cor? rupted. The location of backup superblocks is dependent on the file system's blocksize, the number of blocks per group, and features such as sparse_super.

Additional backup superblocks can be determined by using the mke2fs program using the -n option to print out where the superblocks exist, supposing mke2fs is sup? plied with arguments that are consistent with the file system's layout (e.g. block? size, blocks per group, sparse_super, etc.).

If an alternative superblock is specified and the file system is not opened read-only, e2fsck will make sure that the primary superblock is updated appropriately upon completion of the file system check.

-B blocksize

Normally, e2fsck will search for the superblock at various different block sizes in an attempt to find the appropriate block size. This search can be fooled in some cases. This option forces e2fsck to only try locating the superblock at a particu? lar blocksize. If the superblock is not found, e2fsck will terminate with a fatal error.

-c This option causes e2fsck to use badblocks(8) program to do a read-only scan of the device in order to find any bad blocks. If any bad blocks are found, they are added to the bad block inode to prevent them from being allocated to a file or di? rectory. If this option is specified twice, then the bad block scan will be done using a non-destructive read-write test.

-C fd This option causes e2fsck to write completion information to the specified file de? scriptor so that the progress of the file system check can be monitored. This op? tion is typically used by programs which are running e2fsck. If the file descrip? tor number is negative, then absolute value of the file descriptor will be used, and the progress information will be suppressed initially. It can later be enabled

by sending the e2fsck process a SIGUSR1 signal. If the file descriptor specified is 0, e2fsck will print a completion bar as it goes about its business. This re? quires that e2fsck is running on a video console or terminal.

-d Print debugging output (useless unless you are debugging e2fsck).

-D Optimize directories in file system. This option causes e2fsck to try to optimize all directories, either by re-indexing them if the file system supports directory indexing, or by sorting and compressing directories for smaller directories, or for file systems using traditional linear directories.

Even without the -D option, e2fsck may sometimes optimize a few directories --- for example, if directory indexing is enabled and a directory is not indexed and would benefit from being indexed, or if the index structures are corrupted and need to be rebuilt. The -D option forces all directories in the file system to be optimized.

This can sometimes make them a little smaller and slightly faster to search, but in practice, you should rarely need to use this option.

The -D option will detect directory entries with duplicate names in a single direc? tory, which e2fsck normally does not enforce for performance reasons.

-E extended_options

Set e2fsck extended options. Extended options are comma separated, and may take an argument using the equals ('=') sign. The following options are supported:

ea_ver=extended_attribute_version

Set the version of the extended attribute blocks which e2fsck will re? quire while checking the file system. The version number may be 1 or 2. The default extended attribute version format is 2.

journal_only

Only replay the journal if required, but do not perform any further checks or repairs.

fragcheck

During pass 1, print a detailed report of any discontiguous blocks for files in the file system.

discard

Attempt to discard free blocks and unused inode blocks after the full file system check (discarding blocks is useful on solid state devices and sparse / thin-provisioned storage). Note that discard is done in

pass 5 AFTER the file system has been fully checked and only if it does not contain recognizable errors. However there might be cases where e2fsck does not fully recognize a problem and hence in this case this option may prevent you from further manual data recovery.

nodiscard

Do not attempt to discard free blocks and unused inode blocks. This op? tion is exactly the opposite of discard option. This is set as default.

no_optimize_extents

Do not offer to optimize the extent tree by eliminating unnecessary width or depth. This can also be enabled in the options section of /etc/e2fsck.conf.

optimize_extents

Offer to optimize the extent tree by eliminating unnecessary width or depth. This is the default unless otherwise specified in /etc/e2fsck.conf.

inode_count_fullmap

Trade off using memory for speed when checking a file system with a large number of hard-linked files. The amount of memory required is proportional to the number of inodes in the file system. For large file systems, this can be gigabytes of memory. (For example, a 40TB file system with 2.8 billion inodes will consume an additional 5.7 GB memory if this optimization is enabled.) This optimization can also be enabled in the options section of /etc/e2fsck.conf.

no_inode_count_fullmap

Disable the inode_count_fullmap optimization. This is the default un? less otherwise specified in /etc/e2fsck.conf.

readahead_kb

Use this many KiB of memory to pre-fetch metadata in the hopes of re? ducing e2fsck runtime. By default, this is set to the size of two block groups' inode tables (typically 4MiB on a regular ext4 file sys? tem); if this amount is more than 1/50th of total physical memory, readahead is disabled. Set this to zero to disable readahead entirely.

bmap2extent

Convert block-mapped files to extent-mapped files.

fixes_only

Only fix damaged metadata; do not optimize htree directories or com? press extent trees. This option is incompatible with the -D and -E bmap2extent options.

check_encoding

Force verification of encoded filenames in case-insensitive directo? ries. This is the default mode if the file system has the strict flag enabled.

unshare_blocks

If the file system has shared blocks, with the shared blocks read-only feature enabled, then this will unshare all shared blocks and unset the read-only feature bit. If there is not enough free space then the oper? ation will fail. If the file system does not have the read-only fea? ture bit, but has shared blocks anyway, then this option will have no effect. Note when using this option, if there is no free space to clone blocks, there is no prompt to delete files and instead the operation will fail.

Note that unshare_blocks implies the "-f" option to ensure that all passes are run. Additionally, if "-n" is also specified, e2fsck will simulate trying to allocate enough space to deduplicate. If this fails, the exit code will be non-zero.

-f Force checking even if the file system seems clean.

-F Flush the file system device's buffer caches before beginning. Only really useful for doing e2fsck time trials.

-j external-journal

Set the pathname where the external-journal for this file system can be found.

-k When combined with the -c option, any existing bad blocks in the bad blocks list are preserved, and any new bad blocks found by running badblocks(8) will be added to the existing bad blocks list.

-l filename

Add the block numbers listed in the file specified by filename to the list of bad blocks. The format of this file is the same as the one generated by the bad?

blocks(8) program. Note that the block numbers are based on the blocksize of the file system. Hence, badblocks(8) must be given the blocksize of the file system in order to obtain correct results. As a result, it is much simpler and safer to use the -c option to e2fsck, since it will assure that the correct parameters are passed to the badblocks program.

-L filename

Set the bad blocks list to be the list of blocks specified by filename. (This op? tion is the same as the -l option, except the bad blocks list is cleared before the blocks listed in the file are added to the bad blocks list.)

-n Open the file system read-only, and assume an answer of `no' to all questions. Al? lows e2fsck to be used non-interactively. This option may not be specified at the same time as the -p or -y options.

-p Automatically repair ("preen") the file system. This option will cause e2fsck to automatically fix any file system problems that can be safely fixed without human intervention. If e2fsck discovers a problem which may require the system adminis? trator to take additional corrective action, e2fsck will print a description of the problem and then exit with the value 4 logically or'ed into the exit code. (See the EXIT CODE section.) This option is normally used by the system's boot scripts. It may not be specified at the same time as the -n or -y options.

-r This option does nothing at all; it is provided only for backwards compatibility.

-t Print timing statistics for e2fsck. If this option is used twice, additional tim? ing statistics are printed on a pass by pass basis.

-v Verbose mode.

-V Print version information and exit.

-y Assume an answer of `yes' to all questions; allows e2fsck to be used non-interac? tively. This option may not be specified at the same time as the -n or -p options.

-z undo_file

Before overwriting a file system block, write the old contents of the block to an undo file. This undo file can be used with e2undo(8) to restore the old contents of the file system should something go wrong. If the empty string is passed as the undo_file argument, the undo file will be written to a file named e2fsck-de? vice.e2undo in the directory specified via the E2FSPROGS_UNDO_DIR environment vari? able.

WARNING: The undo file cannot be used to recover from a power or system crash.

EXIT CODE

The exit code returned by e2fsck is the sum of the following conditions:

0 - No errors

1 - File system errors corrected

2 - File system errors corrected, system should be rebooted

4 - File system errors left uncorrected

8 - Operational error

16 - Usage or syntax error

32 - E2fsck canceled by user request

128 - Shared library error

SIGNALS

The following signals have the following effect when sent to e2fsck.

SIGUSR1

This signal causes e2fsck to start displaying a completion bar or emitting progress information. (See discussion of the -C option.)

SIGUSR2

This signal causes e2fsck to stop displaying a completion bar or emitting progress information.

REPORTING BUGS

Almost any piece of software will have bugs. If you manage to find a file system which causes e2fsck to crash, or which e2fsck is unable to repair, please report it to the au? thor.

Please include as much information as possible in your bug report. Ideally, include a complete transcript of the e2fsck run, so I can see exactly what error messages are dis? played. (Make sure the messages printed by e2fsck are in English; if your system has been configured so that e2fsck's messages have been translated into another language, please set the the LC_ALL environment variable to C so that the transcript of e2fsck's output will be useful to me.) If you have a writable file system where the transcript can be stored, the script(1) program is a handy way to save the output of e2fsck to a file.

It is also useful to send the output of dumpe2fs(8). If a specific inode or inodes seems to be giving e2fsck trouble, try running the debugfs(8) command and send the output of the

stat(1u) command run on the relevant inode(s). If the inode is a directory, the debugfs dump command will allow you to extract the contents of the directory inode, which can sent to me after being first run through uuencode(1). The most useful data you can send to help reproduce the bug is a compressed raw image dump of the file system, generated using e2image(8). See the e2image(8) man page for more details.

Always include the full version string which e2fsck displays when it is run, so I know which version you are running.

ENVIRONMENT

E2FSCK_CONFIG

Determines the location of the configuration file (see e2fsck.conf(5)).

AUTHOR

This version of e2fsck was written by Theodore Ts'o <tytso@mit.edu>.

SEE ALSO

e2fsck.conf(5), badblocks(8), dumpe2fs(8), debugfs(8), e2image(8), mke2fs(8), tune2fs(8)