Subscriptions | Downloads | Red Hat Console | Get Support

Products & Services   Knowledgebase   What is the meaning of "ext[3/4]_dx_add_entry: Directory index full!"?

What is the meaning of "ext[3/4]_dx_add_entry: Directory index full!"?

⊘ **SOLUTION VERIFIED** - Updated March 28 2023 at 4:39 AM - English ▾

Environment

- Red Hat Enterprise Linux (all versions)
- ext3 or ext4

Issue

- System log file contains the following messages (dependent on filesystem type ext3 or ext4):

```
kernel: EXT3-fs warning (device sdb1): ext3_dx_add_entry: Directory index full!
```

or

```
kernel: EXT4-fs warning (device dm-2): ext4_dx_add_entry: Directory index full!
```

- What does `ext[3/4]_dx_add_entry: Directory index full` mean?

Resolution

There are a number of possible resolutions:

- Determine if a process is creating unexpected or unwanted files or directories.
- Remove unnecessary or unwanted files or directories.
- Reorganize files and directories within the filesystem to reduce the number of entries in each individual directory.
- Use shorter filenames.

- Change the block size of the filesystem. This option requires a **re-format** of the filesystem, since the block size cannot be changed once it is set.
  CAUTION: Before re-formatting the filesystem backup any important files to another device as reformatting will recreate the filesystem metadata and not retain information about existing files. By default the ext3 block size for small volumes is 1024. If possible format the disk with 4096 bytes (4k) block size as follows:

```
# mkfs -t ext3 -b 4096 /dev/device
```

- Disable `dir_index` option of the filesystem.
  Note: There is no need to unmount the file system.
  CAUTION: Disabling this option may impact performance on lookups in large directories.

```
# tune2fs -O ^dir_index /dev/device
```

To recreate the `dir_index` :

```
# umount /dev/device
# tune2fs -O dir_index /dev/device
# e2fsck -D /dev/device
```

- Optimize directories in filesystem.

```
$ MANWIDTH=80 man e2fsck
   -D      Optimize  directories  in filesystem.  This option causes e2fsck
           to try to optimize all directories, either by reindexing them if
           the  filesystem  supports directory indexing,  or by sorting and
           compressing directories for smaller directories, or for filesys-
           tems 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 filesystem
           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  directory, which e2fsck normally does not enforce
           for performance reasons.
```

- To identify the directory, use the following `auditd` rule. You can get the audit log of the rule with `ausearch -m syscall -k for_ext3_dx_add_entry_warning` .

```
-a exit,always -F arch=b64 -S open -S openat -S creat -S link -S symlink -S
mkdir -F exit=-ENOSPC -k for_ext3_dx_add_entry_warning
```

Root Cause

- The 'directory index full' error will be seen if there are lots of files/directories in a specific directory so that it reaches its indexing limits and cannot keep track further.

- The directory index is an additional tree structure which helps directory lookups, improving performance for huge directories.

- There is a limit in ext3 and ext4 of the directory structure.

    - A directory on ext3 can have at most 31998 sub directories, because an inode can have at most 32000 links. This is one cause of the warning.
    - A directory on ext4 can have at most 64000 sub directories.

- The size of each section of a directory index is limited by the filesystem's block size. If very long filenames are used, fewer entries will fit in the block, leading to 'directory index full' errors earlier than they would occur with shorter filenames. This can become a bigger problem when the filesystem's block size is small (1024-byte or 2048-byte blocks), but will occur with 4096-byte blocks as well.

- For further information about directory indexing, you may refer to the following page : State of the Art: Where we are with the Ext3 filesystem

Diagnostic Steps

Example repoducer

This reproducer illustrates the problem on a filesystem with a blocksize of 1024.

```
[root@rhel8 ~]# uname -r
4.18.0-425.3.1.el8.x86_64
[root@rhel8 ~]# rpm -q e2fsprogs
e2fsprogs-1.45.6-5.el8.x86_64
[root@rhel8 ~]#

[root@rhel8 ~]# truncate -s5G /tmp/ext.image
[root@rhel8 ~]# mkfs.ext4 -b 1024 /tmp/ext.image
mke2fs 1.45.6 (20-Mar-2020)
Discarding device blocks: done
Creating filesystem with 5242880 1k blocks and 327680 inodes
Filesystem UUID: 526bff79-f5ae-47c1-af13-4694e0341b8b
Superblock backups stored on blocks:
        8193, 24577, 40961, 57345, 73729, 204801, 221185, 401409, 663553,
        1024001, 1990657, 2809857, 5120001

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

[root@rhel8 ~]# mount /tmp/ext.image /mnt

[root@rhel8 ~]# for I in $(seq 0 300000); do touch /mnt/file_$I; done
touch: cannot touch '/mnt/file_161845': No space left on device
touch: cannot touch '/mnt/file_164116': No space left on device
touch: cannot touch '/mnt/file_164819': No space left on device
touch: cannot touch '/mnt/file_175308': No space left on device
...
[root@rhel8 ~]# df -i /mnt
Filesystem      Inodes  IUsed IFree IUse% Mounted on
/dev/loop0      327680 299358 28322   92% /mnt
[root@rhel8 ~]#

[root@rhel8 ~]# dmesg | tail
EXT4-fs warning (device loop0): ext4_dx_add_entry:2281: Directory (ino: 2) index
full, reach max htree level :2
EXT4-fs warning (device loop0): ext4_dx_add_entry:2285: Large directory feature
is not enabled on this filesystem
EXT4-fs warning (device loop0): ext4_dx_add_entry:2281: Directory (ino: 2) index
full, reach max htree level :2
EXT4-fs warning (device loop0): ext4_dx_add_entry:2285: Large directory feature
is not enabled on this filesystem
```

Note: Files can still be successfully created if the hash of their filename does not fall into an already full hash bucket.

```
[root@rhel8 ~]# touch /mnt/newfile1
[root@rhel8 ~]# touch /mnt/newfile2
[root@rhel8 ~]# touch /mnt/newfile3
[root@rhel8 ~]# touch /mnt/file_161845
touch: cannot touch '/mnt/file_161845': No space left on device
[root@rhel8 ~]# touch /mnt/newfile4
[root@rhel8 ~]# touch /mnt/newfile5
[root@rhel8 ~]#
```

The htree structure can be inspected using debugfs.

```
[root@rhel8 ~]# umount /mnt
[root@rhel8 ~]# debugfs /tmp/ext.image
debugfs 1.45.6 (20-Mar-2020)
debugfs:  htree /
Root node dump:
         Reserved zero: 0
         Hash Version: 1
         Info length: 8
         Indirect levels: 1
         Flags: 0
Number of entries (count): 123
Number of entries (limit): 123
Checksum: 0x1eb549c9
Entry #0: Hash 0x00000000, block 124
Entry #1: Hash 0x020818ba, block 8347
Entry #2: Hash 0x03fef34c, block 4238
Entry #3: Hash 0x05f14b4a, block 8032
...
Entry #122: Hash 0xfe0b5880, block 8484

Entry #0: Hash 0x00000000, block 124
Number of entries (count): 126
Number of entries (limit): 126
Checksum: 0x7d289d82
Entry #0: Hash 0x00000000, block 1
Entry #1: Hash 0x0002deec, block 14606
Entry #2: Hash 0x000545f4, block 5833
```

Product(s)　**Red Hat Enterprise Linux**　　　　**Component**　**kernel**　　　**Category**　**Learn more**

Tags　**ext3**　**ext4**　**mkfs**　**rhel_4**　**rhel_5**　**rhel_6**　**rhel_7**

This solution is part of Red Hat's fast-track publication program, providing a huge library of solutions that Red Hat engineers have created while supporting our customers. To give you the knowledge you need the instant it becomes available, these articles may be presented in a raw and unedited form.

Comments

Login to see comments

Quick Links

Help

Site Info

Related Sites

About

Red Hat Subscription Value

About Red Hat

Red Hat Jobs

About Red Hat

Jobs

Events

Locations

Contact Red Hat

Red Hat Blog

Diversity, equity, and inclusion

Cool Stuff Store

Red Hat Summit

Copyright © 2024 Red Hat, Inc.

Privacy statement

Terms of use

All policies and guidelines

Digital accessibility

Cookie Preferences and Do Not Sell or Share My Personal Information