

**NAME**

e2fsck.conf – Configuration file for e2fsck

**DESCRIPTION**

*e2fsck.conf* is the configuration file for **e2fsck**(8). It controls the default behavior of **e2fsck**(8) while it is checking ext2 or ext3 filesystems.

The *e2fsck.conf* file uses an INI-style format. Stanzas, or top-level sections, are delimited by square braces: [ ]. Within each section, each line defines a relation, which assigns tags to values, or to a subsection, which contains further relations or subsections. An example of the INI-style format used by this configuration file follows below:

```
[section1]
    tag1 = value_a
    tag1 = value_b
    tag2 = value_c

[section 2]
    tag3 = {
        subtag1 = subtag_value_a
        subtag1 = subtag_value_b
        subtag2 = subtag_value_c
    }
    tag1 = value_d
    tag2 = value_e
}
```

Comments are delimited by a semicolon (;) or a hash (#) character at the beginning of the comment, and are terminated by the end of line character.

Tags and values must be quoted using double quotes if they contain spaces. Within a quoted string, the standard backslash interpretations apply: "\n" (for the newline character), "\t" (for the tab character), "\b" (for the backspace character), and "\\" (for the backslash character).

The following stanzas are used in the *e2fsck.conf* file. They will be described in more detail in future sections of this document.

*[options]*

This stanza contains general configuration parameters for **e2fsck**'s behavior.

*[problems]*

This stanza allows the administrator to reconfigure how e2fsck handles various filesystem inconsistencies.

*[scratch\_files]*

This stanza controls when e2fsck will attempt to use scratch files to reduce the need for memory.

**THE [options] STANZA**

The following relations are defined in the *[options]* stanza.

*allow\_cancellation*

If this relation is set to a boolean value of true, then if the user interrupts e2fsck using ^C, and the filesystem is not explicitly flagged as containing errors, e2fsck will exit with an exit status of 0 instead of 32. This setting defaults to false.

*accept\_time\_fudge*

Unfortunately, due to Windows' unfortunate design decision to configure the hardware clock to tick localtime, instead of the more proper and less error-prone UTC time, many users end up in the situation where the system clock is incorrectly set at the time when e2fsck is run.

Historically this was usually due to some distributions having buggy init scripts and/or installers that didn't correctly detect this case and take appropriate countermeasures. However, it's still possible, despite the best efforts of init script and installer authors to not be able to detect this

misconfiguration, usually due to a buggy or misconfigured virtualization manager or the installer not having access to a network time server during the installation process. So by default, we allow the superblock times to be fudged by up to 24 hours. This can be disabled by setting *accept\_time\_fudge* to the boolean value of false. This setting defaults to true.

#### *broken\_system\_clock*

The **e2fsck(8)** program has some heuristics that assume that the system clock is correct. In addition, many system programs make similar assumptions. For example, the UUID library depends on time not going backwards in order for it to be able to make its guarantees about issuing universally unique ID's. Systems with broken system clocks, are well, broken. However, broken system clocks, particularly in embedded systems, do exist. E2fsck will attempt to use heuristics to determine if the time can not be trusted; and to skip time-based checks if this is true. If this boolean is set to true, then e2fsck will always assume that the system clock can not be trusted.

#### *clear\_test\_fs\_flag*

This boolean relation controls whether or not **e2fsck(8)** will offer to clear the test\_fs flag if the ext4 filesystem is available on the system. It defaults to true.

#### *defer\_check\_on\_battery*

This boolean relation controls whether or not the interval between filesystem checks (either based on time or number of mounts) should be doubled if the system is running on battery. This setting defaults to true.

#### *indexed\_dir\_slack\_percentage*

When **e2fsck(8)** repacks a indexed directory, reserve the specified percentage of empty space in each leaf nodes so that a few new entries can be added to the directory without splitting leaf nodes, so that the average fill ratio of directories can be maintained at a higher, more efficient level. This relation defaults to 20 percent.

## THE **[problems]** STANZA

Each tag in the *[problems]* stanza names a problem code specified with a leading "0x" followed by six hex digits. The value of the tag is a subsection where the relations in that subsection override the default treatment of that particular problem code.

Note that inappropriate settings in this stanza may cause **e2fsck** to behave incorrectly, or even crash. Most system administrators should not be making changes to this section without referring to source code.

Within each problem code's subsection, the following tags may be used:

#### *description*

This relation allows the message which is printed when this filesystem inconsistency is detected to be overridden.

#### *preen\_ok*

This boolean relation overrides the default behavior controlling whether this filesystem problem should be automatically fixed when **e2fsck** is running in preen mode.

*no\_ok* This boolean relation overrides the default behavior determining whether or not the filesystem will be marked as inconsistent if the user declines to fix the reported problem.

#### *no\_default*

This boolean relation overrides whether the default answer for this problem (or question) should be "no".

#### *preen\_nomessage*

This boolean relation overrides the default behavior controlling whether or not the description for this filesystem problem should be suppressed when **e2fsck** is running in preen mode.

#### *no\_nomsg*

This boolean relation overrides the default behavior controlling whether or not the description for this filesystem problem should be suppressed when **e2fsck** is run with the **-n** option.

## THE `[scratch_files]` STANZA

The following relations are defined in the `[scratch_files]` stanza.

### *directory*

If the directory named by this relation exists and is writeable, then e2fsck will attempt to use this directory to store scratch files instead of using in-memory data structures.

### *numdirs\_threshold*

If this relation is set, then in-memory data structures be used if the number of directories in the filesystem are fewer than amount specified.

*dirinfo* This relation controls whether or not the scratch file directory is used instead of an in-memory data structure for directory information. It defaults to true.

*icount* This relation controls whether or not the scratch file directory is used instead of an in-memory data structure when tracking inode counts. It defaults to true.

## EXAMPLES

The following recipe will prevent e2fsck from aborting during the boot process when a filesystem contains orphaned files. (Of course, this is not always a good idea, since critical files that are needed for the security of the system could potentially end up in lost+found, and starting the system without first having a system administrator check things out may be dangerous.)

```
[problems]
    0x040002 = {
        preen_ok = true
        description = "@u @i %i. "
    }
```

## FILES

`/etc/e2fsck.conf`

The configuration file for **e2fsck**(8).

## SEE ALSO

**e2fsck**(8)