NAME

mke2fs.conf − Configuration file for mke2fs

DESCRIPTION

mke2fs.conf is the configuration file for **mke2fs**(8). It controls the default parameters used by **mke2fs**(8) when it is creating ext2 or ext3 filesystems.

The *mke2fs.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 *mke2fs.conf* file. They will be described in more detail in future sections of this document.

[defaults]

Contains relations which define the default parameters used by **mke2fs**(8). In general, these defaults may be overridden by a definition in the **fs_types** stanza, or by an command-line option provided by the user.

[fs_types]

Contains relations which define defaults that should be used for specific filesystem types. The filesystem type can be specified explicitly using the **-T** option to **mke2fs**(8).

THE [defaults] STANZA

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

base_features

This relation specifies the filesystems features which are enabled in newly created filesystems. It may be overridden by the *base_features* relation found in the filesystem or usage type subsection of the *[fs_types]* stanza.

default_features

This relation specifies a set of features that should be added or removed to the features listed in the *base_features* relation. It may be overridden by the filesystem-specific *default_features* in the filesystem or usage type subsection of *[fs_types]*, and by the **-O** command-line option to **mke2fs**(8).

enable_periodic_fsck

This relation specifies whether periodic filesystem checks should be enforced at boot time. If enabled, checks will be forced every 180 days, or after a random number of mounts. These values may be changed later via the **-i** and **-c** command-line options to **tune2fs**(8).

force_undo

This relation, if set to a boolean value of true, forces **mke2fs** to always try to create an undo file, even if the undo file might be huge and it might extend the time to create the filesystem image because the inode table isn't being initialized lazily.

fs_type This relation specifies the default filesystem type if the user does not specify it via the **−t** option, or if **mke2fs** is not started using a program name of the form **mkfs.** *fs-type*. If both the user and the **mke2fs.conf** file does not specify a default filesystem type, mke2fs will use a default filesystem type of *ext3* if a journal was requested via a command-line option, or *ext2* if not.

blocksize

This relation specifies the default blocksize if the user does not specify a blocksize on the command line, and the filesystem-type specific section of the configuration file does not specify a blocksize.

hash_alg

This relation specifies the default hash algorithm used for the new filesystems with hashed b-tree directories. Valid algorithms accepted are: *legacy*, *half_md4*, and *tea*.

inode_ratio

This relation specifies the default inode ratio if the user does not specify one on the command line, and the filesystem-type specific section of the configuration file does not specify a default inode ratio.

inode_size

This relation specifies the default inode size if the user does not specify one on the command line, and the filesystem-type specific section of the configuration file does not specify a default inode size.

undo_dir

This relation specifies the directory where the undo file should be stored. It can be overridden via the **E2FSPROGS_UNDO_DIR** environment variable. If the directory location is set to the value *none*, **mke2fs** will not create an undo file.

THE [fs_types] STANZA

Each tag in the *[fs_types]* stanza names a filesystem type or usage type which can be specified via the **−t** or **−T** options to **mke2fs**(8), respectively.

The **mke2fs** program constructs a list of fs_types by concatenating the filesystem type (i.e., ext2, ext3, etc.) with the usage type list. For most configuration options, **mke2fs** will look for a subsection in the *[fs_types]* stanza corresponding with each entry in the constructed list, with later entries overriding earlier filesystem or usage types. For example, consider the following **mke2fs.conf** fragment:

```
[defaults]
        base_features = sparse_super,filetype,resize_inode,dir_index
        blocksize = 4096
        inode_size = 256
        inode_ratio = 16384

[fs_types]
        ext3 = {
                features = has_journal
        }
        ext4 = {
                features = extents,flex_bg
                inode_size = 256
        }
        small = {
                blocksize = 1024
                inode_ratio = 4096
        }
        floppy = {
                features = ^resize_inode
                blocksize = 1024
                inode_size = 128
        }
```

If mke2fs started with a program name of **mke2fs.ext4**, then the filesystem type of ext4 will be used. If the filesystem is smaller than 3 megabytes, and no usage type is specified, then **mke2fs** will use a default usage type of *floppy*. This results in an fs_types list of "ext4, floppy". Both the ext4 subsection and the floppy subsection define an *inode_size* relation, but since the later entries in the fs_types list supersede earlier ones, the configuration parameter for fs_types.floppy.inode_size will be used, so the filesystem will have an inode size of 128.

The exception to this resolution is the *features* tag, which is specifies a set of changes to the features used by the filesystem, and which is cumulative. So in the above example, first the configuration relation defaults.base_features would enable an initial feature set with the sparse_super, filetype, resize_inode, and dir_index features enabled. Then configuration relation fs_types.ext4.features would enable the extents and flex_bg features, and finally the configuration relation fs_types.floppy.features would remove the resize_inode feature, resulting in a filesystem feature set consisting of the sparse_super, filetype, resize_inode, dir_index, extents_and flex_bg features.

For each filesystem type, the following tags may be used in that fs_type's subsection:

base_features

This relation specifies the features which are initially enabled for this filesystem type. Only one *base_features* will be used, so if there are multiple entries in the fs_types list whose subsections define the *base_features* relation, only the last will be used by **mke2fs**(8).

features

This relation specifies a comma-separated list of features edit requests which modify the feature set used by the newly constructed filesystem. The syntax is the same as the **-O** command-line option to **mke2fs**(8); that is, a feature can be prefixed by a caret ('^') symbol to disable a named feature. Each *feature* relation specified in the fs_types list will be applied in the order found in the fs_types list.

default_features

This relation specifies set of features which should be enabled or disabled after applying the features listed in the *base_features* and *features* relations. It may be overridden by the **-O** command-line option to **mke2fs**(8).

blocksize

This relation specifies the default blocksize if the user does not specify a blocksize on the command line.

lazy_itable_init

This relation is a boolean which specifies whether the inode table should be lazily initialized. It only has meaning if the uninit_bg feature is enabled. If lazy_itable_init is true and the uninit_bg feature is enabled, the inode table will not fully initialized by **mke2fs**(8). This speeds up filesystem initialization noticeably, but it requires the kernel to finish initializing the filesystem in the background when the filesystem is first mounted.

inode_ratio

This relation specifies the default inode ratio if the user does not specify one on the command line.

inode_size
This relation specifies the default inode size if the user does not specify one on the command line.

hash_alg
This relation specifies the default hash algorithm used for the new filesystems with hashed b-tree directories. Valid algorithms accepted are: *legacy*, *half_md4*, and *tea*.

flex_bg_size
This relation specifies the number of block groups that will be packed together to create one large virtual block group on an ext4 filesystem. This improves meta-data locality and performance on meta-data heavy workloads. The number of groups must be a power of 2 and may only be specified if the flex_bg filesystem feature is enabled. *options* This relation specifies additional extended options which should be treated by **mke2fs**(8) as if they were prepended to the argument of the **-E** option. This can be used to configure the default extended options used by **mke2fs**(8) on a per-filesystem type basis.

discard This relation is a boolean which specifies whether the **mke2fs**(8) should attempt to discard device prior to filesystem creation.

FILES

/etc/mke2fs.conf
The configuration file for **mke2fs**(8).

SEE ALSO

mke2fs(8)