NAME

mtree — format of mtree dir hierarchy files

DESCRIPTION

The **mtree** format is a textual format that describes a collection of filesystem objects. Such files are typically used to create or verify directory hierarchies.

General Format

An **mtree** file consists of a series of lines, each providing information about a single filesystem object. Leading whitespace is always ignored.

When encoding file or pathnames, any backslash character or character outside of the 95 printable ASCII characters must be encoded as a backslash followed by three octal digits. When reading mtree files, any appearance of a backslash followed by three octal digits should be converted into the corresponding character.

Each line is interpreted independently as one of the following types:

Blank — Blank lines are ignored.

Comment — Lines beginning with **#** are ignored.

Special — Lines beginning with **/** are special commands that influence the interpretation of later lines.

Relative — If the first whitespace-delimited word has no **/** characters, it is the name of a file in the current directory. Any relative entry that describes a directory changes the current directory.

dot-dot — As a special case, a relative entry with the filename **..** changes the current directory to the parent directory. Options on dot-dot entries are always ignored.

Full — If the first whitespace-delimited word has a **/** character after the first character, it is the pathname of a file relative to the starting directory. There can be multiple full entries describing the same file.

Some tools that process **mtree** files may require that multiple lines describing the same file occur consecutively. It is not permitted for the same file to be mentioned using both a relative and a full file specification.

Special commands

Two special commands are currently defined:

/set — This command defines default values for one or more keywords. It is followed on the same line by one or more whitespace-separated keyword definitions. These definitions apply to all following files that do not specify a value for that keyword.

/unset — This command removes any default value set by a previous **/set** command. It is followed on the same line by one or more keywords separated by whitespace.

Keywords

After the filename, a full or relative entry consists of zero or more whitespace-separated keyword definitions. Each such definition consists of a key from the following list immediately followed by an '=' sign and a value. Software programs reading mtree files should warn about unrecognized keywords.

Currently supported keywords are as follows:

cksum — The checksum of the file using the default algorithm specified by the `cksum`(1) utility.

device — The device number for **block** or **char** file types. The value must be one of the following forms:

format,*major*,*minor*[,*subunit*]
A device with *major*, *minor* and optional *subunit* fields. Their meaning is specified by the operating's system *format*. See below for valid formats.

number
Opaque number (as stored on the file system).

The following values for *format* are recognized: **native**, **386bsd**, **4bsd**, **bsdos**, **freebsd**, **hpux**, **isc**, **linux**, **netbsd**, **osf1**, **sco**, **solaris**, **sunos**, **svr3**, **svr4**, and **ultrix**.

See mknod(8) for more details.

contents The full pathname of a file that holds the contents of this file.

flags The file flags as a symbolic name. See chflags(1) for information on these names. If no flags are to be set the string "none" may be used to override the current default.

gid The file group as a numeric value.

gname The file group as a symbolic name.

ignore Ignore any file hierarchy below this file.

inode The inode number.

link The target of the symbolic link when type=link.

md5 The MD5 message digest of the file.

md5digest
A synonym for **md5**.

mode The current file's permissions as a numeric (octal) or symbolic value.

nlink The number of hard links the file is expected to have.

nochange Make sure this file or directory exists but otherwise ignore all attributes.

optional The file is optional; do not complain about the file if it is not in the file hierarchy.

resdevice
The "resident" device number of the file, e.g. the ID of the device that contains the file. Its format is the same as the one for **device**.

ripemd160digest
The RIPEMD160 message digest of the file.

rmd160 A synonym for **ripemd160digest**.

rmd160digest
A synonym for **ripemd160digest**.

sha1 The FIPS 160-1 ("SHA-1") message digest of the file.

sha1digest
A synonym for **sha1**.

sha256 The FIPS 180-2 ("SHA-256") message digest of the file.

sha256digest
A synonym for **sha256**.

sha384 The FIPS 180-2 ("SHA-384") message digest of the file.

sha384digest
A synonym for **sha384**.

sha512 The FIPS 180-2 ("SHA-512") message digest of the file.

sha512digest
A synonym for **sha512**.

size The size, in bytes, of the file.

time The last modification time of the file.

type The type of the file; may be set to any one of the following:

block	block special device
char	character special device
dir	directory
fifo	fifo
file	regular file
link	symbolic link
socket	socket

uid The file owner as a numeric value.

uname The file owner as a symbolic name.

SEE ALSO

`cksum`(1), `find`(1), `mtree`(8)

HISTORY

The **mtree** utility appeared in 4.3BSD–Reno. The MD5 digest capability was added in FreeBSD 2.1, in response to the widespread use of programs which can spoof `cksum`(1). The SHA-1 and RIPEMD160 digests were added in FreeBSD 4.0, as new attacks have demonstrated weaknesses in MD5. The SHA-256 digest was added in FreeBSD 6.0. Support for file flags was added in FreeBSD 4.0, and mostly comes from NetBSD. The "full" entry format was added by NetBSD.