Rocky Enterprise Linux 9.2 Manual Pages on command 'attr.1'

$ man attr.1

ATTR(1)   XFS Compatibility API   ATTR(1)

NAME

    attr - extended attributes on XFS filesystem objects

SYNOPSIS

    attr [-LRSq] -s attrname [-V attrvalue] pathname

    attr [-LRSq] -g attrname pathname

    attr [-LRSq] -r attrname pathname

    attr [-LRSq] -l pathname

OVERVIEW

    Extended attributes implement the ability for a user to attach

    name:value pairs to objects within the XFS filesystem.

    This document describes the attr command, which is mostly compatible

    with the IRIX command of the same name. It is thus aimed specifically

    at users of the XFS filesystem - for filesystem independent extended

    attribute manipulation, consult the getfattr(1) and setfattr(1) docu?

    mentation.

    Extended attributes can be used to store meta-information about the

    file. For example "character-set=kanji" could tell a document browser

to use the Kanji character set when displaying that document and "thumbnail=..." could provide a reduced resolution overview of a high resolution graphic image.

In the XFS filesystem, the names can be up to 256 bytes in length, ter? minated by the first 0 byte. The intent is that they be printable ASCII (or other character set) names for the attribute. The values can be up to 64KB of arbitrary binary data.

Attributes can be attached to all types of XFS inodes: regular files, directories, symbolic links, device nodes, etc.

XFS uses 2 disjoint attribute name spaces associated with every filesystem object. They are the root and user address spaces. The root address space is accessible only to the superuser, and then only by specifying a flag argument to the function call. Other users will not see or be able to modify attributes in the root address space. The user address space is protected by the normal file permissions mecha? nism, so the owner of the file can decide who is able to see and/or modify the value of attributes on any particular file.

DESCRIPTION

The attr utility allows the manipulation of extended attributes associ? ated with filesystem objects from within shell scripts.

There are four main operations that attr can perform:

GET   The -g attrname option tells attr to search the named object and print (to stdout) the value associated with that attribute name. With the -q flag, stdout will be exactly and only the value of the attribute, suitable for storage directly into a file or pro? cessing via a piped command.

LIST   The -l option tells attr to list the names of all the attributes that are associated with the object, and the number of bytes in the value of each of those attributes. With the -q flag, stdout will be a simple list of only the attribute names, one per line, suitable for input into a script.

REMOVE The -r attrname option tells attr to remove an attribute with the given name from the object if the attribute exists. There

is no output on successful completion.

SET/CREATE

> The -s attrname option tells attr to set the named attribute of
> the object to the value read from stdin. If an attribute with
> that name already exists, its value will be replaced with this
> one. If an attribute with that name does not already exist, one
> will be created with this value. With the -V attrvalue flag,
> the attribute will be set to have a value of attrvalue and stdin
> will not be read. With the -q flag, stdout will not be used.
> Without the -q flag, a message showing the attribute name and
> the entire value will be printed.

When the -L option is given and the named object is a symbolic link,
operate on the attributes of the object referenced by the symbolic
link. Without this option, operate on the attributes of the symbolic
link itself.

When the -R option is given and the process has appropriate privileges,
operate in the root attribute namespace rather that the USER attribute
namespace.

The -S option is similar, except it specifies use of the security at‐
tribute namespace.

When the -q option is given attr will try to keep quiet. It will out‐
put error messages (to stderr) but will not print status messages (to
stdout).

NOTES

The standard file interchange/archive programs tar(1), and cpio(1) will
not archive or restore extended attributes, while the xfsdump(8) pro‐
gram will.

CAVEATS

The list option present in the IRIX version of this command is not sup‐
ported. getfattr provides a mechanism to retrieve all of the attribute
names.

AUTHOR

Andreas Gruenbacher, <andreas.gruenbacher@gmail.com> and the SGI XFS

development team, <linux-xfs@oss.sgi.com>.

Please send your bug reports or comments to <https://savan?
nah.nongnu.org/bugs/?group=attr> or <acl-devel@nongnu.org>.

SEE ALSO

getfattr(1), setfattr(1), attr_get(3), attr_set(3), attr_multi(3),

attr_remove(3), attr(5), xfsdump(8)