



Full credit is given to the above companies including the OS that this PDF file was generated!

Linux Ubuntu 26.04 Manual Pages on command 'fgetxattr.2'

\$ man fgetxattr.2

getxattr(2) System Calls Manual getxattr(2)

NAME

getxattr, lgetxattr, fgetxattr - retrieve an extended attribute value

LIBRARY

Standard C library (libc, -lc)

SYNOPSIS

```
#include <sys/xattr.h>
```

```
ssize_t getxattr(size_t size;
                 const char *path, const char *name,
                 void value[size], size_t size);
```

```
ssize_t lgetxattr(size_t size;
                  const char *path, const char *name,
                  void value[size], size_t size);
```

```
ssize_t fgetxattr(size_t size;
                  int fd, const char *name,
                  void value[size], size_t size);
```

DESCRIPTION

Extended attributes are name:value pairs associated with inodes (files, directories, symbolic links, etc.). They are extensions to the normal attributes which are associated with all inodes in the system (i.e., the stat(2) data). A complete overview of extended attributes concepts can be found in xattr(7).

getxattr() retrieves the value of the extended attribute identified by name and associated with the given path in the filesystem. The attribute value is placed in the buffer pointed

to by value; size specifies the size of that buffer. The return value of the call is the number of bytes placed in value.

lgetxattr() is identical to getxattr(), except in the case of a symbolic link, where the link itself is interrogated, not the file that it refers to.

fgetxattr() is identical to getxattr(), only the open file referred to by fd (as returned by open(2)) is interrogated in place of path.

An extended attribute name is a null-terminated string. The name includes a namespace prefix; there may be several, disjoint namespaces associated with an individual inode. The value of an extended attribute is a chunk of arbitrary textual or binary data that was assigned using setxattr(2).

If size is specified as zero, these calls return the current size of the named extended attribute (and leave value unchanged). This can be used to determine the size of the buffer that should be supplied in a subsequent call. (But, bear in mind that there is a possibility that the attribute value may change between the two calls, so that it is still necessary to check the return status from the second call.)

RETURN VALUE

On success, these calls return a nonnegative value which is the size (in bytes) of the extended attribute value. On failure, -1 is returned and errno is set to indicate the error.

ERRORS

E2BIG The size of the attribute value is larger than the maximum size allowed; the attribute cannot be retrieved. This can happen on filesystems that support very large attribute values such as NFSv4, for example.

ENODATA

The named attribute does not exist, or the process has no access to this attribute.

ENOTSUP

Extended attributes are not supported by the filesystem, or are disabled.

ERANGE The size of the value buffer is too small to hold the result.

In addition, the errors documented in stat(2) can also occur.

STANDARDS

Linux.

HISTORY

Linux 2.4, glibc 2.3.

EXAMPLES

See listxattr(2).

SEE ALSO

getfattr(1), setfattr(1), listxattr(2), open(2), removexattr(2), setxattr(2), stat(2), sym?

link(7), xattr(7)

Linux man-pages 6.17

2026-02-08

getxattr(2)