

**NAME**

**acl\_get\_fd**, **acl\_get\_fd\_np**, **acl\_get\_file**, **acl\_get\_link\_np** - get an ACL for a file

**LIBRARY**

Standard C Library (libc, -lc)

**SYNOPSIS**

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

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

```
acl_t
```

```
acl_get_fd(int fd);
```

```
acl_t
```

```
acl_get_fd_np(int fd, acl_type_t type);
```

```
acl_t
```

```
acl_get_file(const char *path_p, acl_type_t type);
```

```
acl_t
```

```
acl_get_link_np(const char *path_p, acl_type_t type);
```

**DESCRIPTION**

The **acl\_get\_fd()**, **acl\_get\_file()**, **acl\_get\_link\_np()**, and **acl\_get\_fd\_np()** each allow the retrieval of an ACL from a file. The **acl\_get\_fd()** is a POSIX.1e call that allows the retrieval of an ACL of type `ACL_TYPE_ACCESS` from a file descriptor. The **acl\_get\_fd\_np()** function is a non-portable form of **acl\_get\_fd()** that allows the retrieval of any type of ACL from a file descriptor. The **acl\_get\_file()** function is a POSIX.1e call that allows the retrieval of a specified type of ACL from a file by name; **acl\_get\_link\_np()** is a non-portable variation on **acl\_get\_file()** which does not follow a symlink if the target of the call is a symlink.

These functions may cause memory to be allocated. The caller should free any releasable memory, when the new ACL is no longer required, by calling **acl\_free(3)** with the *(void \*)acl\_t* as an argument.

The ACL in the working storage is an independent copy of the ACL associated with the object referred to by *fd*. The ACL in the working storage shall not participate in any access control decisions.

Valid values for the *type* argument are:

`ACL_TYPE_ACCESS`     POSIX.1e access ACL

ACL\_TYPE\_DEFAULT    POSIX.1e default ACL  
ACL\_TYPE\_NFS4        NFSv4 ACL

The ACL returned will be branded accordingly.

## IMPLEMENTATION NOTES

FreeBSD's support for POSIX.1e interfaces and features is still under development at this time.

## RETURN VALUES

Upon successful completion, the function shall return a pointer to the ACL that was retrieved. Otherwise, a value of *(acl\_t)NULL* shall be returned, and *errno* shall be set to indicate the error.

## ERRORS

If any of the following conditions occur, the **acl\_get\_fd()** function shall return a value of *(acl\_t)NULL* and set *errno* to the corresponding value:

- [EACCES]            Search permission is denied for a component of the path prefix, or the object exists and the process does not have appropriate access rights.
- [EBADF]            The *fd* argument is not a valid file descriptor.
- [EINVAL]           The ACL type passed is invalid for this file object.
- [ENAMETOOLONG]    A component of a pathname exceeded 255 characters, or an entire path name exceeded 1023 characters.
- [ENOENT]           The named object does not exist, or the *path\_p* argument points to an empty string.
- [ENOMEM]           Insufficient memory available to fulfill request.
- [EOPNOTSUPP]      The file system does not support ACL retrieval.

## SEE ALSO

[acl\(3\)](#), [acl\\_free\(3\)](#), [acl\\_get\(3\)](#), [acl\\_get\\_brand\\_np\(3\)](#), [acl\\_set\(3\)](#), [posix1e\(3\)](#)

## STANDARDS

POSIX.1e is described in IEEE POSIX.1e draft 17. Discussion of the draft continues on the cross-platform POSIX.1e implementation mailing list. To join this list, see the FreeBSD POSIX.1e

implementation page for more information.

**HISTORY**

POSIX.1e support was introduced in FreeBSD 4.0, and development continues.

**AUTHORS**

Robert N M Watson