

**NAME**

**acl\_set\_fd**, **acl\_set\_fd\_np**, **acl\_set\_file**, **acl\_set\_link\_np** - set an ACL for a file

**LIBRARY**

Standard C Library (libc, -lc)

**SYNOPSIS**

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

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

*int*

```
acl_set_fd(int fd, acl_t acl);
```

*int*

```
acl_set_fd_np(int fd, acl_t acl, acl_type_t type);
```

*int*

```
acl_set_file(const char *path_p, acl_type_t type, acl_t acl);
```

*int*

```
acl_set_link_np(const char *path_p, acl_type_t type, acl_t acl);
```

**DESCRIPTION**

The **acl\_set\_fd()**, **acl\_set\_fd\_np()**, **acl\_set\_file()**, and **acl\_set\_link\_np()** each associate an ACL with an object referred to by *fd* or *path\_p*. The **acl\_set\_fd\_np()** and **acl\_set\_link\_np()** functions are not POSIX.1e calls. The **acl\_set\_fd()** function allows only the setting of ACLs of type `ACL_TYPE_ACCESS` whereas **acl\_set\_fd\_np()** allows the setting of ACLs of any type. The **acl\_set\_link\_np()** function acts on a symlink rather than its target, if the target of the path is a symlink.

Valid values for the *type* argument are:

<code>ACL_TYPE_ACCESS</code>	POSIX.1e access ACL
<code>ACL_TYPE_DEFAULT</code>	POSIX.1e default ACL
<code>ACL_TYPE_NFS4</code>	NFSv4 ACL

Trying to set `ACL_TYPE_NFS4` with *acl* branded as POSIX.1e, or `ACL_TYPE_ACCESS` or `ACL_TYPE_DEFAULT` with ACL branded as NFSv4, will result in error.

**IMPLEMENTATION NOTES**

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

## RETURN VALUES

Upon successful completion, the value 0 is returned; otherwise the value -1 is returned and the global variable *errno* is set to indicate the error.

## ERRORS

If any of the following conditions occur, these functions shall return -1 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 <i>fd</i> argument is not a valid file descriptor.   |
| [EINVAL]       | Argument <i>acl</i> does not point to a valid ACL for this object, or the ACL type specified in <i>type</i> is invalid for this object, or there is branding mismatch. |
| [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 <i>path_p</i> argument points to an empty string.  |
| [ENOMEM]       | Insufficient memory available to fulfill request.  |
| [ENOSPC]       | The directory or file system that would contain the new ACL cannot be extended, or the file system is out of file allocation resources.                                |
| [EOPNOTSUPP]   | The file system does not support ACL retrieval.  |
| [EROFS]        | This function requires modification of a file system which is currently read-only.   |

## SEE ALSO

[acl\(3\)](#), [acl\\_delete\(3\)](#), [acl\\_get\(3\)](#), [acl\\_get\\_brand\\_np\(3\)](#), [acl\\_valid\(3\)](#), [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