

**NAME**

**acl\_valid**, **acl\_valid\_fd\_np**, **acl\_valid\_file\_np**, **acl\_valid\_link\_np** - validate an ACL

**LIBRARY**

Standard C Library (libc, -lc)

**SYNOPSIS**

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

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

*int*

```
acl_valid(acl_t acl);
```

*int*

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

*int*

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

*int*

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

**DESCRIPTION**

These functions check that the ACL referred to by the argument *acl* is valid. The POSIX.1e routine, **acl\_valid()**, checks this validity only with POSIX.1e ACL semantics, and irrespective of the context in which the ACL is to be used. The non-portable forms, **acl\_valid\_fd\_np()**, **acl\_valid\_file\_np()**, and **acl\_valid\_link\_np()** allow an ACL to be checked in the context of a specific acl type, *type*, and file system object. In environments where additional ACL types are supported than just POSIX.1e, this makes more sense. Whereas **acl\_valid\_file\_np()** will follow the symlink if the specified path is to a symlink, **acl\_valid\_link\_np()** will not.

For POSIX.1e semantics, the checks include:

- The three required entries (ACL\_USER\_OBJ, ACL\_GROUP\_OBJ, and ACL\_OTHER) shall exist exactly once in the ACL. If the ACL contains any ACL\_USER, ACL\_GROUP, or any other implementation-defined entries in the file group class then one ACL\_MASK entry shall also be required. The ACL shall contain at most one ACL\_MASK entry.
- The qualifier field shall be unique among all entries of the same POSIX.1e ACL facility defined tag type. The tag type field shall contain valid values including any implementation-defined values.

Validation of the values of the qualifier field is implementation-defined.

The POSIX.1e **acl\_valid()** function may reorder the ACL for the purposes of verification; the non-portable validation functions will not.

## 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 *fd* argument is not a valid file descriptor.

[EINVAL]           Argument *acl* does not point to a valid ACL.

One or more of the required ACL entries is not present in *acl*.

The ACL contains entries that are not unique.

The file system rejects the ACL based on fs-specific semantics issues.

[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\_get(3), acl\_init(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