NAME

acl_extended_file_np, **acl_extended_file_nofollow_np**, **acl_extended_link_np** - checks if the file has extended ACLs set

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

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

int
acl_extended_file_np(*const char* path_p*);

int
acl_extended_file_nofollow_np(*const char* path_p*);

int
acl_extended_link_np(*const char* path_p*);

DESCRIPTION

The **acl_extended_file_np**() function is a non-portable call that checks if the file or directory referred to by the argument *path_p* contains extended access ACLs. The **acl_extended_file_nofollow_np**() function works the same way, except it does not follow symlinks. The **acl_extended_link_np**() function is a synonim to **acl_extended_file_nofollow_np**() named in FreeBSD style. An ACL is considered to be extended access one if it contains entries other than the three required entries of tag types ACL_USER_OBJ, ACL_GROUP_OBJ and ACL_OTHER.

RETURN VALUES

Upon successful completion, this function returns 0 if the file object does not contain extended access ACLs and 1 in the other case. Otherwise, the value -1 is returned, and *errno* indicates the error.

ERRORS

If any of the following conditions occur, the **acl_extended_file_np**() function shall return a value of *-1* and set *errno* to the corresponding value:

[EACCES]	Search permission is denied for a component of the path prefix.

SEE ALSO

extattr_get_file(2), 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

Gleb Popov