

NAME

`acl_to_any_text` — convert an ACL to text

LIBRARY

Linux Access Control Lists library (`libacl`, `-lacl`).

SYNOPSIS

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

char *
acl_to_any_text(acl_t acl, const char *prefix, char separator,
               int options);
```

DESCRIPTION

The **`acl_to_any_text()`** function translates the ACL pointed to by the argument *acl* into a NULL terminated character string. This character string is composed of the ACL entries contained in *acl*, in the entry text format described on *acl(5)*. Entries are separated from each other by the *separator* character. If the argument *prefix* is not `(const char *)NULL`, each entry is prefixed by this character string.

If the argument *options* is 0, ACL entries are converted using the entry tag type keywords **`user`**, **`group`**, **`mask`**, and **`other`**. User IDs and group IDs of ACL entries that contain such qualifiers are converted to their corresponding names; if an identifier has no corresponding name, a decimal number string is produced. The ACL text representation contains no additional comments. A bitwise combinations of the following *options* can be used to modify the result:

TEXT_ABBREVIATE

Instead of the full tag type keywords, single letter abbreviations are used. The abbreviation for **`user`** is **`u`**, the abbreviation for **`group`** is **`g`**, the abbreviation for **`mask`** is **`m`**, and the abbreviation for **`other`** is **`o`**.

TEXT_NUMERIC_IDS

User IDs and group IDs are included as decimal numbers instead of names.

TEXT_SOME_EFFECTIVE

A comment containing the effective permissions of the ACL entry is included after ACL entries that contain permissions which are ineffective because they are masked by an **`ACL_MASK`** entry. The ACL entry and the comment are separated by a tab character.

TEXT_ALL_EFFECTIVE

A comment containing the effective permissions of the ACL entry is included after all ACL entries that are affected by an **`ACL_MASK`** entry. The comment is included even if the permissions contained in the ACL entry equal the effective permissions. The ACL entry and the comment are separated by a tab character.

TEXT_SMART_INDENT

This option is used in combination with the **`TEXT_SOME_EFFECTIVE`** or **`TEXT_ALL_EFFECTIVE`** option. The number of tab characters inserted between the ACL entry and the comment is increased so that the comment is aligned to the fourth tab stop position. A tab width of 8 characters is assumed.

The ACL referred to by *acl* is not changed.

This function allocates any memory necessary to contain the string and returns a pointer to the string. The caller should free any releasable memory, when the new string is no longer required, by calling **`acl_free()`** with the `(void*)char` returned by **`acl_to_any_text()`** as an argument.

RETURN VALUE

On success, this function returns a pointer to the text representation of the ACL. On error, a value of `(char *)NULL` is returned, and *errno* is set appropriately.

ERRORS

If any of the following conditions occur, the **acl_to_any_text()** function returns a value of (char *)NULL and sets *errno* to the corresponding value:

- | | |
|----------|---|
| [EINVAL] | The argument <i>acl</i> is not a valid pointer to an ACL.

The ACL referenced by <i>acl</i> contains one or more improperly formed ACL entries, or for some other reason cannot be translated into the text form of an ACL. |
| [ENOMEM] | The character string to be returned requires more memory than is allowed by the hardware or system-imposed memory management constraints. |

STANDARDS

This is a non-portable, Linux specific extension to the ACL manipulation functions defined in IEEE Std 1003.1e draft 17 (“POSIX.1e”, abandoned).

SEE ALSO

acl_from_text(3), *acl_to_text(3)*, *acl_free(3)*, *acl(5)*

AUTHOR

Written by Andreas Gruenbacher <andreas.gruenbacher@gmail.com>.