

explain\_acl\_get\_fd(3)

explain\_acl\_get\_fd(3)

## NAME

explain\_acl\_get\_fd – explain *acl\_get\_fd*(3) errors

## SYNOPSIS

```
#include <libexplain/acl_get_fd.h>

const char *explain_acl_get_fd(int fildes);
const char *explain_errno_acl_get_fd(int errnum, int fildes);
void explain_message_acl_get_fd(char *message, int message_size, int fildes);
void explain_message_errno_acl_get_fd(char *message, int message_size, int errnum, int fildes);
```

## DESCRIPTION

These functions may be used to obtain explanations for errors returned by the *acl\_get\_fd*(3) system call.

### explain\_acl\_get\_fd

```
const char *explain_acl_get_fd(int fildes);
```

The **explain\_acl\_get\_fd** function is used to obtain an explanation of an error returned by the *acl\_get\_fd*(3) system call. The least the message will contain is the value of `strerror(errno)`, but usually it will do much better, and indicate the underlying cause in more detail.

The *errno* global variable will be used to obtain the error value to be decoded.

*fildes* The original *fildes*, exactly as passed to the *acl\_get\_fd*(3) system call.

Returns: The message explaining the error. This message buffer is shared by all libexplain functions which do not supply a buffer in their argument list. This will be overwritten by the next call to any libexplain function which shares this buffer, including other threads.

**Note:** This function is **not** thread safe, because it shares a return buffer across all threads, and many other functions in this library.

**Example:** This function is intended to be used in a fashion similar to the following example:

```
acl_t result = acl_get_fd(fildes);
if (result < 0)
{
    fprintf(stderr, "%s\n", explain_acl_get_fd(fildes));
    exit(EXIT_FAILURE);
}
```

The above code example is available pre-packaged as the *explain\_acl\_get\_fd\_or\_die*(3) function.

### explain\_errno\_acl\_get\_fd

```
const char *explain_errno_acl_get_fd(int errnum, int fildes);
```

The **explain\_errno\_acl\_get\_fd** function is used to obtain an explanation of an error returned by the *acl\_get\_fd*(3) system call. The least the message will contain is the value of `strerror(errno)`, but usually it will do much better, and indicate the underlying cause in more detail.

*errnum* The error value to be decoded, usually obtained from the *errno* global variable just before this function is called. This is necessary if you need to call **any** code between the system call to be explained and this function, because many libc functions will alter the value of *errno*.

*fildes* The original *fildes*, exactly as passed to the *acl\_get\_fd*(3) system call.

Returns: The message explaining the error. This message buffer is shared by all libexplain functions which do not supply a buffer in their argument list. This will be overwritten by the next call to any libexplain function which shares this buffer, including other threads.

**Note:** This function is **not** thread safe, because it shares a return buffer across all threads, and many other functions in this library.

**Example:** This function is intended to be used in a fashion similar to the following example:

```
acl_t result = acl_get_fd(fildes);
if (result < 0)
{
    int err = errno;
    fprintf(stderr, "%s\n", explain_errno_acl_get_fd(err,
        fildes));
}
```



explain\_acl\_get\_fd(3)

explain\_acl\_get\_fd(3)

```
        exit(EXIT_FAILURE);
    }
```

The above code example is available pre-packaged as the *explain\_acl\_get\_fd\_or\_die(3)* function.

### explain\_message\_acl\_get\_fd

```
void explain_message_acl_get_fd(char *message, int message_size, int fildes);
```

The **explain\_message\_acl\_get\_fd** function is used to obtain an explanation of an error returned by the *acl\_get\_fd(3)* system call. The least the message will contain is the value of `strerror(errno)`, but usually it will do much better, and indicate the underlying cause in more detail.

The *errno* global variable will be used to obtain the error value to be decoded.

*message* The location in which to store the returned message. If a suitable message return buffer is supplied, this function is thread safe.

*message\_size*

The size in bytes of the location in which to store the returned message.

*fildes*

The original *fildes*, exactly as passed to the *acl\_get\_fd(3)* system call.

**Example:** This function is intended to be used in a fashion similar to the following example:

```
acl_t result = acl_get_fd(fildes);
if (result < 0)
{
    char message[3000];
    explain_message_acl_get_fd(message, sizeof(message),
                              fildes);
    fprintf(stderr, "%s\n", message);
    exit(EXIT_FAILURE);
}
```

The above code example is available pre-packaged as the *explain\_acl\_get\_fd\_or\_die(3)* function.

### explain\_message\_errno\_acl\_get\_fd

```
void explain_message_errno_acl_get_fd(char *message, int message_size, int errnum, int fildes);
```

The **explain\_message\_errno\_acl\_get\_fd** function is used to obtain an explanation of an error returned by the *acl\_get\_fd(3)* system call. The least the message will contain is the value of `strerror(errno)`, but usually it will do much better, and indicate the underlying cause in more detail.

*message* The location in which to store the returned message. If a suitable message return buffer is supplied, this function is thread safe.

*message\_size*

The size in bytes of the location in which to store the returned message.

*errnum*

The error value to be decoded, usually obtained from the *errno* global variable just before this function is called. This is necessary if you need to call **any** code between the system call to be explained and this function, because many libc functions will alter the value of *errno*.

*fildes*

The original *fildes*, exactly as passed to the *acl\_get\_fd(3)* system call.

**Example:** This function is intended to be used in a fashion similar to the following example:

```
acl_t result = acl_get_fd(fildes);
if (result < 0)
{
    int err = errno;
    char message[3000];
    explain_message_errno_acl_get_fd(message, sizeof(message),
                                     err, fildes);
    fprintf(stderr, "%s\n", message);
    exit(EXIT_FAILURE);
}
```

The above code example is available pre-packaged as the *explain\_acl\_get\_fd\_or\_die(3)* function.



`explain_acl_get_fd(3)`

`explain_acl_get_fd(3)`

## SEE ALSO

*acl\_get\_fd(3)*

Execute *acl\_get\_fd(3)*

*explain\_acl\_get\_fd\_or\_die(3)*

Execute *acl\_get\_fd(3)* and report errors

## COPYRIGHT

libexplain version 1.4

Copyright © 2013 Peter Miller

