

explain_iconv(3)

explain_iconv(3)

NAMEexplain_iconv – explain *iconv*(3) errors**SYNOPSIS**

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

const char *explain_iconv(iconv_t cd, char **inbuf, size_t *inbytesleft, char **outbuf, size_t *outbytesleft);
const char *explain_errno_iconv(int errnum, iconv_t cd, char **inbuf, size_t *inbytesleft, char **outbuf, size_t *outbytesleft);
void explain_message_iconv(char *message, int message_size, iconv_t cd, char **inbuf, size_t *inbytesleft, char **outbuf, size_t *outbytesleft);
void explain_message_errno_iconv(char *message, int message_size, int errnum, iconv_t cd, char **inbuf, size_t *inbytesleft, char **outbuf, size_t *outbytesleft);
```

DESCRIPTION

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

explain_iconv

```
const char *explain_iconv(iconv_t cd, char **inbuf, size_t *inbytesleft, char **outbuf, size_t *outbytesleft);
```

The **explain_iconv** function is used to obtain an explanation of an error returned by the *iconv*(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.

cd The original *cd*, exactly as passed to the *iconv*(3) system call.

inbuf The original *inbuf*, exactly as passed to the *iconv*(3) system call.

inbytesleft

The original *inbytesleft*, exactly as passed to the *iconv*(3) system call.

outbuf The original *outbuf*, exactly as passed to the *iconv*(3) system call.

outbytesleft

The original *outbytesleft*, exactly as passed to the *iconv*(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:

```
errno = 0;
size_t result = iconv(cd, inbuf, inbytesleft, outbuf, outbytesleft);
if (result < 0 && errno != 0)
{
    fprintf(stderr, "%s\n", explain_iconv(cd, inbuf, inbytesleft, outbuf, outbytesleft));
    exit(EXIT_FAILURE);
}
```

The above code example is available pre-packaged as the *explain_iconv_or_die*(3) function.

explain_errno_iconv

```
const char *explain_errno_iconv(int errnum, iconv_t cd, char **inbuf, size_t *inbytesleft, char **outbuf, size_t *outbytesleft);
```

The **explain_errno_iconv** function is used to obtain an explanation of an error returned by the *iconv*(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



explain_iconv(3)

explain_iconv(3)

to be explained and this function, because many libc functions will alter the value of *errno*.

cd The original *cd*, exactly as passed to the *iconv(3)* system call.

inbuf The original *inbuf*, exactly as passed to the *iconv(3)* system call.

inbytesleft

The original *inbytesleft*, exactly as passed to the *iconv(3)* system call.

outbuf The original *outbuf*, exactly as passed to the *iconv(3)* system call.

outbytesleft

The original *outbytesleft*, exactly as passed to the *iconv(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:

```
errno = 0;
size_t result = iconv(cd, inbuf, inbytesleft, outbuf, out-
bytesleft);
if (result < 0 && errno != 0)
{
    int err = errno;
    fprintf(stderr, "%s\n", explain_errno_iconv(err, cd, inbuf,
inbytesleft, outbuf, outbytesleft));
    exit(EXIT_FAILURE);
}
```

The above code example is available pre-packaged as the *explain_iconv_or_die(3)* function.

explain_message_iconv

```
void explain_message_iconv(char *message, int message_size, iconv_t cd, char **inbuf, size_t
*inbytesleft, char **outbuf, size_t *outbytesleft);
```

The **explain_message_iconv** function is used to obtain an explanation of an error returned by the *iconv(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.

cd The original *cd*, exactly as passed to the *iconv(3)* system call.

inbuf The original *inbuf*, exactly as passed to the *iconv(3)* system call.

inbytesleft

The original *inbytesleft*, exactly as passed to the *iconv(3)* system call.

outbuf The original *outbuf*, exactly as passed to the *iconv(3)* system call.

outbytesleft

The original *outbytesleft*, exactly as passed to the *iconv(3)* system call.

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

```
errno = 0;
size_t result = iconv(cd, inbuf, inbytesleft, outbuf, out-
bytesleft);
if (result < 0 && errno != 0)
{
    char message[3000];
    explain_message_iconv(message, sizeof(message), cd, inbuf,
```



explain_iconv(3)

explain_iconv(3)

```

        inbytesleft, outbuf, outbytesleft);
        fprintf(stderr, "%s\n", message);
        exit(EXIT_FAILURE);
    }

```

The above code example is available pre-packaged as the *explain_iconv_or_die(3)* function.

explain_message_errno_iconv

```
void explain_message_errno_iconv(char *message, int message_size, int errnum, iconv_t cd, char
**inbuf, size_t *inbytesleft, char **outbuf, size_t *outbytesleft);
```

The **explain_message_errno_iconv** function is used to obtain an explanation of an error returned by the *iconv(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*.

cd The original *cd*, exactly as passed to the *iconv(3)* system call.

inbuf The original *inbuf*, exactly as passed to the *iconv(3)* system call.

inbytesleft

The original *inbytesleft*, exactly as passed to the *iconv(3)* system call.

outbuf The original *outbuf*, exactly as passed to the *iconv(3)* system call.

outbytesleft

The original *outbytesleft*, exactly as passed to the *iconv(3)* system call.

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

```

errno = 0;
size_t result = iconv(cd, inbuf, inbytesleft, outbuf, out-
bytesleft);
if (result < 0 && errno != 0)
{
    int err = errno;
    char message[3000];
    explain_message_errno_iconv(message, sizeof(message), err,
cd, inbuf, inbytesleft, outbuf, outbytesleft);
    fprintf(stderr, "%s\n", message);
    exit(EXIT_FAILURE);
}

```

The above code example is available pre-packaged as the *explain_iconv_or_die(3)* function.

SEE ALSO

iconv(3) perform character set conversion

explain_iconv_or_die(3)

perform character set conversion and report errors

COPYRIGHT

libexplain version 1.4

Copyright © 2013 Peter Miller

