

NAME

iconv - perform character set conversion

SYNOPSIS

```
#include <iconv.h>

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

DESCRIPTION

The argument *cd* must be a conversion descriptor created using the function **iconv_open**.

The main case is when *inbuf* is not NULL and **inbuf* is not NULL. In this case, the **iconv** function converts the multibyte sequence starting at **inbuf* to a multibyte sequence starting at **outbuf*. At most **inbytesleft* bytes, starting at **inbuf*, will be read. At most **outbytesleft* bytes, starting at **outbuf*, will be written.

The **iconv** function converts one multibyte character at a time, and for each character conversion it increments **inbuf* and decrements **inbytesleft* by the number of converted input bytes, it increments **outbuf* and decrements **outbytesleft* by the number of converted output bytes, and it updates the conversion state contained in *cd*. If the character encoding of the input is stateful, the **iconv** function can also convert a sequence of input bytes to an update of the conversion state without producing any output bytes; such input is called a *shift sequence*. The conversion can stop for four reasons:

1. An invalid multibyte sequence is encountered in the input. In this case it sets **errno** to **EILSEQ** and returns (size_t)(-1). **inbuf* is left pointing to the beginning of the invalid multibyte sequence.
2. The input byte sequence has been entirely converted, i.e. **inbytesleft* has gone down to 0. In this case **iconv** returns the number of non-reversible conversions performed during this call.
3. An incomplete multibyte sequence is encountered in the input, and the input byte sequence terminates after it. In this case it sets **errno** to **EINVAL** and returns (size_t)(-1). **inbuf* is left pointing to the beginning of the incomplete multibyte sequence.
4. The output buffer has no more room for the next converted character. In this case it sets **errno** to **E2BIG** and returns (size_t)(-1).

A different case is when *inbuf* is NULL or **inbuf* is NULL, but *outbuf* is not NULL and **outbuf* is not NULL. In this case, the **iconv** function attempts to set *cd*'s conversion state to the initial state and store a corresponding shift sequence at **outbuf*. At most **outbytesleft* bytes, starting at **outbuf*, will be written. If the output buffer has no more room for this reset sequence, it sets **errno** to **E2BIG** and returns (size_t)(-1). Otherwise it increments **outbuf* and decrements **outbytesleft* by the number of bytes written.

A third case is when *inbuf* is NULL or **inbuf* is NULL, and *outbuf* is NULL or **outbuf* is NULL. In this case, the **iconv** function sets *cd*'s conversion state to the initial state.

RETURN VALUE

The **iconv** function returns the number of characters converted in a non-reversible way during this call; reversible conversions are not counted. In case of error, it sets **errno** and returns (size_t)(-1).

ERRORS

The following errors can occur, among others:

E2BIG

There is not sufficient room at **outbuf*.

EILSEQ

An invalid multibyte sequence has been encountered in the input.

EINVAL

An incomplete multibyte sequence has been encountered in the input.

CONFORMING TO

POSIX:2001

SEE ALSO

[iconv_open\(3\)](#), [iconvctl\(3\)](#) [iconv_close\(3\)](#)