NAME

iconv – code conversion function

SYNOPSIS

#include <iconv.h>

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

MT-LEVEL

MT-Safe

DESCRIPTION

The **iconv( )** function converts the sequence of characters from one codeset, in the array specified by *inbuf*, into a sequence of corresponding characters in another codeset, in the array specified by *outbuf*. The codesets are those specified in the *iconv_open*( ) call that returned the conversion descriptor, *cd*. The *inbuf* argument points to a variable that points to the first character in the input buffer and *inbytesleft* indicates the number of bytes to the end of the buffer to be converted. The *outbuf* argument points to a variable that points to the first available byte in the output buffer and *outbytesleft* indicates the number of the available bytes to the end of the buffer.

For state-dependent encodings, the conversion descriptor *cd* is placed into its initial shift state by a call for which *inbuf* is a null pointer, or for which *inbuf* points to a null pointer. When **iconv( )** is called in this way, and if *outbuf* is not a null pointer or a pointer to a null pointer, and *outbytesleft* points to a positive value, **iconv( )** will place, into the output buffer, the byte sequence to change the output buffer to its initial shift state. If the output buffer is not large enough to hold the entire reset sequence, **iconv( )** will fail and set **errno** to **E2BIG**. Subsequent calls with *inbuf* as other than a null pointer or a pointer to a null pointer cause the conversion to take place from the current state of the conversion descriptor.

If a sequence of input bytes does not form a valid character in the specified codeset, conversion stops after the previous successfully converted character. If the input buffer ends with an incomplete character or shift sequence, conversion stops after the previous successfully converted bytes. If the output buffer is not large enough to hold the entire converted input, conversion stops just prior to the input bytes that would cause the output buffer to overflow. The variable pointed to by *inbuf* is updated to point to the byte following the last byte successfully used in the conversion. The value pointed to by *inbytesleft* is decremented to reflect the number of bytes still not converted in the input buffer. The variable pointed to by *outbuf* is updated to point to the byte following the last byte of converted output data. The value pointed to by *outbytesleft* is decremented to reflect the number of bytes still available in the output buffer. For state-dependent encodings, the conversion descriptor is updated to reflect the shift state in effect at the end of the last successfully converted byte sequence.

If **iconv( )** encounters a character in the input buffer that is legal, but for which an identical character does not exist in the target codeset, **iconv( )** performs an implementation-defined conversion on this character.

RETURN VALUES

The **iconv( )** function updates the variables pointed to by the arguments to reflect the extent of the conversion and returns the number of non-identical conversions performed. If the entire string in the input buffer is converted, the value pointed to by *inbytesleft* will be **0**. If the input conversion is stopped due to any conditions mentioned above, the value pointed to by *inbytesleft* will be non-zero and **errno** is set to indicate the condition. If an error occurs **iconv( )** returns **(size_t) −1** and sets **errno** to indicate the error.

ERRORS

The **iconv( )** function will fail if:

EILSEQ Input conversion stopped due to an input byte that does not belong to the input codeset.

E2BIG Input conversion stopped due to lack of space in the output buffer.

EINVAL Input conversion stopped due to an incomplete character or shift sequence at the end of the input buffer.

The **iconv( )** function may fail if:

EBADF The *cd* argument is not a valid open conversion descriptor.

FILES

/usr/lib/iconv/*.so conversion modules

SEE ALSO

iconv(1), **iconv_open**(3), **iconv_close**(3), **iconv**(5)