NAME

libunicode - UTF-8 to UTF-32 conversions and various operations

SYNOPSIS

#include <unicode.h>

size_t
uni8_encode(*uint8_t *dst, size_t dstsz, uint32_t point*);

size_t
uni8_decode(*const uint8_t *src, uint32_t *point*);

size_t
uni8_sizeof(*uint8_t c*);

size_t
uni8_length(*const uint8_t *src*);

size_t
uni8_to32(*const uint8_t *src, uint32_t *dst, size_t dstsz*);

size_t
uni32_sizeof(*uint32_t point*);

size_t
uni32_length(*const uint32_t *src*);

size_t
uni32_requires(*const uint32_t *src*);

size_t
uni32_to8(*const uint32_t *src, uint8_t *dst, size_t dstsz*);

int
uni_isalpha(*uint32_t c*);

int
uni_iscontrol(*uint32_t c*);

int

uni_isdigit(*uint32_t c*);

int
uni_islower(*uint32_t c*);

int
uni_isspace(*uint32_t c*);

int
uni_istitle(*uint32_t c*);

int
uni_isupper(*uint32_t c*);

uint32_t
uni_toupper(*uint32_t c*);

uint32_t
uni_tolower(*uint32_t c*);

DESCRIPTION

This set of functions allows back-and-forth conversions between UTF-8 and UTF-32 character sets. All input strings (both UTF-8 and UTF-32) are considered to be NUL-terminated when use as input. Output strings are always NUL terminated unless specified otherwise.

Functions prefixed with "uni8_" are referring to UTF-8 source or destination.

Functions prefixed with "uni32_" are analogous to their respective "uni8_" counterparts if applicable.

Finally, generic functions prefixed with "uni_" do not perform any conversion and are made for character class classificiation.

The **uni8_encode**() function transforms the unicode character *point* and store the result as UTF-8 string into *dst* of *dstsz* bytes long. The output string is *not* NUL terminated and must be at least 4 bytes long, otherwise it may be truncated.

The **uni8_decode**() function reads the UTF-8 NUL-terminated *src* input string and converts the result into *point* as unicode character.

The **uni8_sizeof**() function returns the number of bytes that are following the byte *c* in a multibytes sequence. It can be used while iterating a UTF-8 string to jump a specific number of bytes while encountering a multibytes sequence.

The **uni8_length**() function returns the number of unicode characters (which is lesser or equal of the number of bytes) in the UTF-8 NUL terminated *src* string.

The **uni8_to32**() function converts the UTF-8 *src* input string into the *dst* array of *dstsz* bytes long. The function writes at most *dstsz* bytes including the NUL terminator character, make sure to reserve an additional space for it before calling this function.

The **uni32_sizeof**() function returns the number of UTF-8 characters required to convert the unicode *point* character to UTF-8.

The **uni32_length**() function returns the number of unicode characters present in the UTF-32 NUL terminated *src* string.

The **uni32_requires**() function computes the total number of bytes (excluding the NUL terminator) that are required to build a UTF-8 string from the UTF-32 NUL terminated *src* string.

The **uni32_to8**() function converts the UTF-32 NUL terminated *src* string and stores the result as UTF-8 in *dst* of *dstsz* bytes long. The function writes at most *dstsz* bytes including the NUL terminator character, make sure to reserve an additional space for it before calling this function.

The **uni_isalpha**() returns non-zero if the the unicode character *c* is considered alphanumeric class.

The **uni_iscontrol**() returns non-zero if the unicode character *c* is considered as a control character class.

The **uni_isdigit**() returns non-zero if the the unicode character *c* is considered numeric class.

The **uni_islower**() returns non-zero if the the unicode character *c* is considered lower case class.

The **uni_istitle**() returns non-zero if the the unicode character *c* is considered title case class.

The **uni_isupper**() returns non-zero if the the unicode character *c* is considered upper case class.

The **uni_toupper**() returns the upper case variant of the unicode character *c*.

The **uni_tolower**() returns the lower case variant of the unicode character *c*.

RETURN VALUES

The **uni8_encode**(), **uni8_to32**() and **uni32_to8**() functions return the number of bytes written (excluding NUL terminator) into *dst* or -1 in case of error.

The **uni8_decode**() function returns the number of bytes parsed from *src* string or -1 in case of error.

The **uni8_sizeof**() and **uni8_length**() functions return -1 in case of error.

The **uni32_sizeof**() and **uni32_requires**() functions return -1 in case of error.

ERRORS

The global *errno* variable can be set in case of error using the following macro constants:

[EILSEQ]	When an invalid sequence was found. Can be set in **uni8_decode**(), **uni8_sizeof**(), **uni8_length**(), **uni8_to32**(), **uni32_sizeof**(), **uni32_requires**() and **uni32_to8**().
[ERANGE]	Set when there wasn't enough room to store conversion. Can be set in **uni8_encode**(), **uni8_to32**(), **uni32_requires**() and **uni32_to8**().