

NAME

punycode_encode - API function

SYNOPSIS

```
#include <punycode.h>
```

```
int punycode_encode(size_t input_length, const punycode_uint [] input, const unsigned char [] case_flags, size_t * output_length, char [] output);
```

ARGUMENTS

size_t input_length

The number of code points in the *input* array and the number of flags in the *case_flags* array.

const punycode_uint [] input

An array of code points. They are presumed to be Unicode code points, but that is not strictly REQUIRED. The array contains code points, not code units. UTF-16 uses code units D800 through DFFF to refer to code points 10000..10FFFF. The code points D800..DFFF do not occur in any valid Unicode string. The code points that can occur in Unicode strings (0..D7FF and E000..10FFFF) are also called Unicode scalar values.

const unsigned char [] case_flags

A **NULL** pointer or an array of boolean values parallel to the *input* array. Nonzero (true, flagged) suggests that the corresponding Unicode character be forced to uppercase after being decoded (if possible), and zero (false, unflagged) suggests that it be forced to lowercase (if possible). ASCII code points (0..7F) are encoded literally, except that ASCII letters are forced to uppercase or lowercase according to the corresponding case flags. If *case_flags* is a **NULL** pointer then ASCII letters are left as they are, and other code points are treated as unflagged.

size_t * output_length

The caller passes in the maximum number of ASCII code points that it can receive. On successful return it will contain the number of ASCII code points actually output.

char [] output An array of ASCII code points. It is *not* null-terminated; it will contain zeros if and only if the *input* contains zeros. (Of course the caller can leave room for a terminator and add one if needed.)

DESCRIPTION

Converts a sequence of code points (presumed to be Unicode code points) to Punycode.

Return value: The return value can be any of the **Punycode_status** values defined above except **PUNYCODE_BAD_INPUT**. If not **PUNYCODE_SUCCESS**, then *output_size* and *output* might contain garbage.

DESCRIPTION

Converts a sequence of code points (presumed to be Unicode code points) to Punycode.

Return value: The return value can be any of the **Punycode_status** values defined above except **PUNYCODE_BAD_INPUT**. If not **PUNYCODE_SUCCESS**, then *output_size* and *output* might contain garbage.

REPORTING BUGS

Report bugs to <help-libidn@gnu.org>.

General guidelines for reporting bugs: <http://www.gnu.org/gethelp/>

GNU Libidn home page: <http://www.gnu.org/software/libidn/>

COPYRIGHT

Copyright (C) 2002-2021 Simon Josefsson.

Copying and distribution of this file, with or without modification, are permitted in any medium without royalty provided the copyright notice and this notice are preserved.

SEE ALSO

The full documentation for **libidn** is maintained as a Texinfo manual. If the **info** and **libidn** programs are properly installed at your site, the command

info libidn

should give you access to the complete manual. As an alternative you may obtain the manual from:

<http://www.gnu.org/software/libidn/manual/>