

unicode.h(3)

libmtp

unicode.h(3)

NAME

libmtp –

SYNOPSIS**Functions**

```
int ucs2_strlen (uint16_t const *const)
char * utf16_to_utf8 (LIBMTP_mtpdevice_t *, const uint16_t *)
uint16_t * utf8_to_utf16 (LIBMTP_mtpdevice_t *, const char *)
void strip_7bit_from_utf8 (char *str)
```

Detailed Description

This file contains general Unicode string manipulation functions. It mainly consist of functions for converting between UCS-2 (used on the devices) and UTF-8 (used by several applications).

For a deeper understanding of Unicode encoding formats see the Wikipedia entries for UTF-16/UCS-2 and UTF-8.

Copyright (C) 2005-2007 Linus Walleij triad AT df DOT lth DOT se

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

Function Documentation

void strip_7bit_from_utf8 (char * str)

This helper function simply removes any consecutive chars

0x7F and replace them with an underscore. In UTF-8

consecutive chars > 0x7F represent one single character so it has to be done like this (and it's elegant).

It will only shrink the string in size so no copying is needed.

int ucs2_strlen (uint16_t const *const unicstr)

Gets the length (in characters, not bytes) of a unicode UCS-2 string, eg a string which physically is 0x00 0x41 0x00 0x00 will return a value of 1.

Parameters:

unicstr a UCS-2 Unicode string

Returns:

the length of the string, in number of characters. If you want to know the length in bytes, multiply this by two and add two (for zero terminator).

Referenced by utf16_to_utf8(), and utf8_to_utf16().

char* utf16_to_utf8 (LIBMTP_mtpdevice_t * device, const uint16_t * unicstr)

Converts a big-endian UTF-16 2-byte string to a UTF-8 string. Actually just a UCS-2 internal conversion routine that strips off the BOM if there is one.

Parameters:

device a pointer to the current device.

unicstr the UTF-16 unicode string to convert

Returns:

a UTF-8 string.

References LIBMTP_mtpdevice_struct::params, STRING_BUFFER_LENGTH, and ucs2_strlen().

uint16_t* utf8_to_utf16 (LIBMTP_mtpdevice_t * device, const char * localstr)

Converts a UTF-8 string to a big-endian UTF-16 2-byte string. Actually just a UCS-2 internal conversion.

Parameters:

unicode.h(3)

libmtp

unicode.h(3)

device a pointer to the current device.

localstr the UTF-8 unicode string to convert

Returns:

a UTF-16 string.

References LIBMTP_mtpdevice_struct::params, STRING_BUFFER_LENGTH, and ucs2_strlen().

Author

Generated automatically by Doxygen for libmtp from the source code.

