



Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'OPENSSL_buf2hexstr.3oss1' command

\$ man OPENSSL_buf2hexstr.3oss1

OPENSSL_HEXCHAR2INT(3oss1) OpenSSL OPENSSL_HEXCHAR2INT(3oss1)

NAME

OPENSSL_hexchar2int, OPENSSL_hexstr2buf_ex, OPENSSL_hexstr2buf,
OPENSSL_buf2hexstr_ex, OPENSSL_buf2hexstr - Hex encoding and decoding
functions

SYNOPSIS

```
#include <openssl/crypto.h>
```

```
int OPENSSL_hexchar2int(unsigned char c);  
int OPENSSL_hexstr2buf_ex(unsigned char *buf, size_t buf_n, long *buflen,  
                          const char *str, const char sep);  
unsigned char *OPENSSL_hexstr2buf(const char *str, long *len);  
int OPENSSL_buf2hexstr_ex(char *str, size_t str_n, size_t *strlength,  
                          const unsigned char *buf, long buflen,  
                          const char sep);  
char *OPENSSL_buf2hexstr(const unsigned char *buf, long buflen);
```

DESCRIPTION

OPENSSL_hexchar2int() converts a hexadecimal character to its numeric equivalent.

`OPENSSL_hexstr2buf_ex()` decodes the hex string `str` and places the resulting string of bytes in the given `buf`. The character `sep` is the separator between the bytes, setting this to `'\0'` means that there is no separator. `buf_n` gives the size of the buffer. If `buflen` is not `NULL`, it is filled in with the result length. To find out how large the result will be, call this function with `NULL` for `buf`. Colons between two-character hex "bytes" are accepted and ignored. An odd number of hex digits is an error.

`OPENSSL_hexstr2buf()` does the same thing as `OPENSSL_hexstr2buf_ex()`, but allocates the space for the result, and returns the result. It uses a default separator of `'.'`. The memory is allocated by calling `OPENSSL_malloc()` and should be released by calling `OPENSSL_free()`.

`OPENSSL_buf2hexstr_ex()` encodes the contents of the given `buf` with length `buflen` and places the resulting hexadecimal character string in the given `str`. The character `sep` is the separator between the bytes, setting this to `'\0'` means that there is no separator. `str_n` gives the size of the of the string buffer. If `strlen` is not `NULL`, it is filled in with the result length. To find out how large the result will be, call this function with `NULL` for `str`.

`OPENSSL_buf2hexstr()` does the same thing as `OPENSSL_buf2hexstr_ex()`, but allocates the space for the result, and returns the result. It uses a default separator of `'.'`. The memory is allocated by calling `OPENSSL_malloc()` and should be released by calling `OPENSSL_free()`.

RETURN VALUES

`OPENSSL_hexchar2int` returns the value of a decoded hex character, or -1 on error.

`OPENSSL_buf2hexstr()` and `OPENSSL_hexstr2buf()` return a pointer to allocated memory, or `NULL` on error.

OPENSSL_buf2hexstr_ex() and OPENSSL_hexstr2buf_ex() return 1 on success, or 0 on error.

COPYRIGHT

Copyright 2016-2022 The OpenSSL Project Authors. All Rights Reserved.

Licensed under the Apache License 2.0 (the "License"). You may not use this file except in compliance with the License. You can obtain a copy in the file LICENSE in the source distribution or at <https://www.openssl.org/source/license.html>.

3.0.7 2023-07-13 OPENSSL_HEXCHAR2INT(3oss)