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 'MD2.3ossl' command

$ man MD2.3ossl

MD5(3ossl) OpenSSL MD5(3ossl)

NAME

MD2, MD4, MD5, MD2_Init, MD2_Update, MD2_Final, MD4_Init, MD4_Update, MD4_Final, MD5_Init, MD5_Update, MD5_Final - MD2, MD4, and MD5 hash functions

SYNOPSIS

```
#include <openssl/md2.h>
```

The following functions have been deprecated since OpenSSL 3.0, and can be hidden entirely by defining OPENSSL_API_COMPAT with a suitable version value, see openssl_user_macros(7):

```
unsigned char *MD2(const unsigned char *d, unsigned long n, unsigned char *md);

int MD2_Init(MD2_CTX *c);
int MD2_Update(MD2_CTX *c, const unsigned char *data, unsigned long len);
int MD2_Final(unsigned char *md, MD2_CTX *c);

#include <openssl/md4.h>
```

The following functions have been deprecated since OpenSSL 3.0, and can

be hidden entirely by defining OPENSSL_API_COMPAT with a suitable
version value, see openssl_user_macros(7):

```
unsigned char *MD4(const unsigned char *d, unsigned long n, unsigned char *md);

int MD4_Init(MD4_CTX *c);
int MD4_Update(MD4_CTX *c, const void *data, unsigned long len);
int MD4_Final(unsigned char *md, MD4_CTX *c);

#include <openssl/md5.h>
```

The following functions have been deprecated since OpenSSL 3.0, and can
be hidden entirely by defining OPENSSL_API_COMPAT with a suitable
version value, see openssl_user_macros(7):

```
unsigned char *MD5(const unsigned char *d, unsigned long n, unsigned char *md);

int MD5_Init(MD5_CTX *c);
int MD5_Update(MD5_CTX *c, const void *data, unsigned long len);
int MD5_Final(unsigned char *md, MD5_CTX *c);
```

DESCRIPTION

All of the functions described on this page are deprecated.
Applications should instead use EVP_DigestInit_ex(3),
EVP_DigestUpdate(3) and EVP_DigestFinal_ex(3).

MD2, MD4, and MD5 are cryptographic hash functions with a 128 bit
output.

MD2(), MD4(), and MD5() compute the MD2, MD4, and MD5 message digest of
the n bytes at d and place it in md (which must have space for
MD2_DIGEST_LENGTH == MD4_DIGEST_LENGTH == MD5_DIGEST_LENGTH == 16 bytes
of output). If md is NULL, the digest is placed in a static array.

The following functions may be used if the message is not completely stored in memory:

MD2_Init() initializes a MD2_CTX structure.

MD2_Update() can be called repeatedly with chunks of the message to be hashed (len bytes at data).

MD2_Final() places the message digest in md, which must have space for MD2_DIGEST_LENGTH == 16 bytes of output, and erases the MD2_CTX.

MD4_Init(), MD4_Update(), MD4_Final(), MD5_Init(), MD5_Update(), and MD5_Final() are analogous using an MD4_CTX and MD5_CTX structure.

Applications should use the higher level functions EVP_DigestInit(3) etc. instead of calling the hash functions directly.

NOTE

MD2, MD4, and MD5 are recommended only for compatibility with existing applications. In new applications, SHA-1 or RIPEMD-160 should be preferred.

RETURN VALUES

MD2(), MD4(), and MD5() return pointers to the hash value.

MD2_Init(), MD2_Update(), MD2_Final(), MD4_Init(), MD4_Update(), MD4_Final(), MD5_Init(), MD5_Update(), and MD5_Final() return 1 for success, 0 otherwise.

CONFORMING TO

RFC 1319, RFC 1320, RFC 1321

SEE ALSO

EVP_DigestInit(3)

HISTORY

All of these functions were deprecated in OpenSSL 3.0.

COPYRIGHT

Copyright 2000-2021 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>.