

**NAME**

MDC2, MDC2\_Init, MDC2\_Update, MDC2\_Final - MDC2 hash function

**SYNOPSIS**

```
#include <openssl/mdc2.h>

unsigned char *MDC2(const unsigned char *d, unsigned long n,
unsigned char *md);

int MDC2_Init(MDC2_CTX *c);
int MDC2_Update(MDC2_CTX *c, const unsigned char *data,
unsigned long len);
int MDC2_Final(unsigned char *md, MDC2_CTX *c);
```

**DESCRIPTION**

MDC2 is a method to construct hash functions with 128 bit output from block ciphers. These functions are an implementation of MDC2 with DES.

*MDC2()* computes the MDC2 message digest of the **n** bytes at **d** and places it in **md** (which must have space for MDC2\_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:

*MDC2\_Init()* initializes a **MDC2\_CTX** structure.

*MDC2\_Update()* can be called repeatedly with chunks of the message to be hashed (**len** bytes at **data**).

*MDC2\_Final()* places the message digest in **md**, which must have space for MDC2\_DIGEST\_LENGTH == 16 bytes of output, and erases the **MDC2\_CTX**.

Applications should use the higher level functions *EVP\_DigestInit(3)* etc. instead of calling the hash functions directly.

**RETURN VALUES**

*MDC2()* returns a pointer to the hash value.

*MDC2\_Init()*, *MDC2\_Update()* and *MDC2\_Final()* return 1 for success, 0 otherwise.

**CONFORMING TO**

ISO/IEC 10118-2, with DES

**SEE ALSO**

*sha(3)*, *EVP\_DigestInit(3)*

**HISTORY**

*MDC2()*, *MDC2\_Init()*, *MDC2\_Update()* and *MDC2\_Final()* are available since SSLeay 0.8.