



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

Rocky Enterprise Linux 9.2 Manual Pages on command 'EVP_rc2_cfb64.3ossl'

\$ man EVP_rc2_cfb64.3ossl

EVP_RC2_CBC(3ossl) OpenSSL EVP_RC2_CBC(3ossl)

NAME

EVP_rc2_cbc, EVP_rc2_cfb, EVP_rc2_cfb64, EVP_rc2_ecb, EVP_rc2_ofb,
EVP_rc2_40_cbc, EVP_rc2_64_cbc - EVP RC2 cipher

SYNOPSIS

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

```
const EVP_CIPHER *EVP_rc2_cbc(void);  
const EVP_CIPHER *EVP_rc2_cfb(void);  
const EVP_CIPHER *EVP_rc2_cfb64(void);  
const EVP_CIPHER *EVP_rc2_ecb(void);  
const EVP_CIPHER *EVP_rc2_ofb(void);  
const EVP_CIPHER *EVP_rc2_40_cbc(void);  
const EVP_CIPHER *EVP_rc2_64_cbc(void);
```

DESCRIPTION

The RC2 encryption algorithm for EVP.

`EVP_rc2_cbc()`, `EVP_rc2_cfb()`, `EVP_rc2_cfb64()`, `EVP_rc2_ecb()`,
`EVP_rc2_ofb()`

RC2 encryption algorithm in CBC, CFB, ECB and OFB modes respectively. This is a variable key length cipher with an additional parameter called "effective key bits" or "effective key length". By default both are set to 128 bits.

`EVP_rc2_40_cbc()`, `EVP_rc2_64_cbc()`

RC2 algorithm in CBC mode with a default key length and effective key length of 40 and 64 bits.

WARNING: these functions are obsolete. Their usage should be replaced with the `EVP_rc2_cbc()`, `EVP_CIPHER_CTX_set_key_length()` and `EVP_CIPHER_CTX_ctrl()` functions to set the key length and effective key length.

RETURN VALUES

These functions return an `EVP_CIPHER` structure that contains the implementation of the symmetric cipher. See `EVP_CIPHER_meth_new(3)` for details of the `EVP_CIPHER` structure.

SEE ALSO

`evp(7)`, `EVP_EncryptInit(3)`, `EVP_CIPHER_meth_new(3)`

COPYRIGHT

Copyright 2017-2020 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

EVP_RC2_CBC(30ssl)