



*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 'EVP\_PKEY\_CTX\_set\_scrypt\_maxmem\_bytes.3ossl' com**

***\$ man EVP\_PKEY\_CTX\_set\_scrypt\_maxmem\_bytes.3ossl***

EVP\_PKEY\_CTX\_SET\_SCRYPT\_N(3ossl) OpenSSL EVP\_PKEY\_CTX\_SET\_SCRYPT\_N(3ossl)

### NAME

EVP\_PKEY\_CTX\_set1\_scrypt\_salt, EVP\_PKEY\_CTX\_set\_scrypt\_N,  
EVP\_PKEY\_CTX\_set\_scrypt\_r, EVP\_PKEY\_CTX\_set\_scrypt\_p,  
EVP\_PKEY\_CTX\_set\_scrypt\_maxmem\_bytes - EVP\_PKEY scrypt KDF support  
functions

### SYNOPSIS

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

```
int EVP_PKEY_CTX_set1_scrypt_salt(EVP_PKEY_CTX *pctx, unsigned char *salt,  
int saltlen);
```

```
int EVP_PKEY_CTX_set_scrypt_N(EVP_PKEY_CTX *pctx, uint64_t N);
```

```
int EVP_PKEY_CTX_set_scrypt_r(EVP_PKEY_CTX *pctx, uint64_t r);
```

```
int EVP_PKEY_CTX_set_scrypt_p(EVP_PKEY_CTX *pctx, uint64_t p);
```

```
int EVP_PKEY_CTX_set_scrypt_maxmem_bytes(EVP_PKEY_CTX *pctx,  
uint64_t maxmem);
```

## DESCRIPTION

These functions are used to set up the necessary data to use the script KDF. For more information on script, see `EVP_KDF-SCRIPT(7)`.

`EVP_PKEY_CTX_set1_script_salt()` sets the saltlen bytes long salt value.

`EVP_PKEY_CTX_set_script_N()`, `EVP_PKEY_CTX_set_script_r()` and `EVP_PKEY_CTX_set_script_p()` configure the work factors N, r and p.

`EVP_PKEY_CTX_set_script_maxmem_bytes()` sets how much RAM key derivation may maximally use, given in bytes. If RAM is exceeded because the load factors are chosen too high, the key derivation will fail.

## STRING CTRLS

script also supports string based control operations via `EVP_PKEY_CTX_ctrl_str(3)`. Similarly, the salt can either be specified using the type parameter "salt" or in hex encoding by using the "hexsalt" parameter. The work factors N, r and p as well as maxmem\_bytes can be set by using the parameters "N", "r", "p" and "maxmem\_bytes", respectively.

## NOTES

There is a newer generic API for KDFs, `EVP_KDF(3)`, which is preferred over the `EVP_PKEY` method.

The script KDF also uses `EVP_PKEY_CTX_set1_pbe_pass()` as well as the value from the string controls "pass" and "hexpass". See `EVP_PKEY_CTX_set1_pbe_pass(3)`.

## RETURN VALUES

All these functions return 1 for success and 0 or a negative value for failure. In particular a return value of -2 indicates the operation is not supported by the public key algorithm.

## SEE ALSO

EVP\_KDF(3) EVP\_PKEY\_CTX\_new(3), EVP\_PKEY\_CTX\_ctrl\_str(3),  
EVP\_PKEY\_derive(3)

## HISTORY

All of the functions described here were converted from macros to  
functions in OpenSSL 3.0.

## 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\_PKEY\_CTX\_SET\_SCRIPT\_N(3ossl)