

**NAME**

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

**SYNOPSIS**

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

```
int EVP_PKEY_CTX_set1_script_salt(EVP_PKEY_CTX *pctx, unsigned char *salt,  

                                  int saltlen);
```

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

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

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

```
int EVP_PKEY_CTX_set_script_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 scrypt 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>.