



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_get_octet_string_param.3oss1' command

`$ man EVP_PKEY_get_octet_string_param.3oss1`

`EVP_PKEY_GETTABLE_PARAMS(3oss1)` OpenSSL `EVP_PKEY_GETTABLE_PARAMS(3oss1)`

NAME

`EVP_PKEY_gettable_params`, `EVP_PKEY_get_params`, `EVP_PKEY_get_int_param`,
`EVP_PKEY_get_size_t_param`, `EVP_PKEY_get_bn_param`,
`EVP_PKEY_get_utf8_string_param`, `EVP_PKEY_get_octet_string_param` -
retrieve key parameters from a key

SYNOPSIS

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

```
const OSSL_PARAM *EVP_PKEY_gettable_params(EVP_PKEY *pkey);
```

```
int EVP_PKEY_get_params(const EVP_PKEY *pkey, OSSL_PARAM params[]);
```

```
int EVP_PKEY_get_int_param(const EVP_PKEY *pkey, const char *key_name,  
                           int *out);
```

```
int EVP_PKEY_get_size_t_param(const EVP_PKEY *pkey, const char *key_name,  
                              size_t *out);
```

```
int EVP_PKEY_get_bn_param(const EVP_PKEY *pkey, const char *key_name,  
                           BIGNUM **bn);
```

```
int EVP_PKEY_get_utf8_string_param(const EVP_PKEY *pkey, const char *key_name,  
                                   char *str, size_t max_buf_sz,  
                                   size_t *out_len);
```

```
int EVP_PKEY_get_octet_string_param(const EVP_PKEY *pkey, const char *key_name,
```

```
unsigned char *buf, size_t max_buf_sz,  
size_t *out_len);
```

DESCRIPTION

`EVP_PKEY_get_params()` retrieves parameters from the key `pkey`, according to the contents of `params`. See `OSSL_PARAM(3)` for information about parameters.

`EVP_PKEY_gettable_params()` returns a constant list of `params` indicating the names and types of key parameters that can be retrieved. See `OSSL_PARAM(3)` for information about parameters.

An `OSSL_PARAM` of type `OSSL_PARAM_INTEGER` or `OSSL_PARAM_UNSIGNED_INTEGER` is of arbitrary length. Such a parameter can be obtained using any of the functions `EVP_PKEY_get_int_param()`, `EVP_PKEY_get_size_t_param()` or `EVP_PKEY_get_bn_param()`. Attempting to obtain an integer value that does not fit into a native C `int` type will cause `EVP_PKEY_get_int_param()` to fail. Similarly attempting to obtain an integer value that is negative or does not fit into a native C `size_t` type using `EVP_PKEY_get_size_t_param()` will also fail.

`EVP_PKEY_get_int_param()` retrieves a key `pkey` integer value `*out` associated with a name of `key_name` if it fits into "int" type. For parameters that do not fit into "int" use `EVP_PKEY_get_bn_param()`.

`EVP_PKEY_get_size_t_param()` retrieves a key `pkey` `size_t` value `*out` associated with a name of `key_name` if it fits into "size_t" type. For parameters that do not fit into "size_t" use `EVP_PKEY_get_bn_param()`.

`EVP_PKEY_get_bn_param()` retrieves a key `pkey` `BIGNUM` value `**bn` associated with a name of `key_name`. If `*bn` is `NULL` then the `BIGNUM` is allocated by the method.

`EVP_PKEY_get_utf8_string_param()` get a key pkey UTF8 string value into a buffer `str` of maximum size `max_buf_sz` associated with a name of `key_name`. The maximum size must be large enough to accomodate the string value including a terminating NUL byte, or this function will fail. If `out_len` is not NULL, `*out_len` is set to the length of the string not including the terminating NUL byte. The required buffer size not including the terminating NUL byte can be obtained from `*out_len` by calling the function with `str` set to NULL.

`EVP_PKEY_get_octet_string_param()` get a key pkey's octet string value into a buffer `buf` of maximum size `max_buf_sz` associated with a name of `key_name`. If `out_len` is not NULL, `*out_len` is set to the length of the contents. The required buffer size can be obtained from `*out_len` by calling the function with `buf` set to NULL.

NOTES

These functions only work for `EVP_PKEY`s that contain a provider side key.

RETURN VALUES

`EVP_PKEY_gettable_params()` returns NULL on error or if it is not supported.

All other methods return 1 if a value associated with the key's `key_name` was successfully returned, or 0 if there was an error. An error may be returned by methods `EVP_PKEY_get_utf8_string_param()` and `EVP_PKEY_get_octet_string_param()` if `max_buf_sz` is not big enough to hold the value. If `out_len` is not NULL, `*out_len` will be assigned the required buffer size to hold the value.

EXAMPLES

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

```

char curve_name[64];
unsigned char pub[256];
BIGNUM *bn_priv = NULL;

/*
 * NB: assumes 'key' is set up before the next step. In this example the key
 * is an EC key.
 */

if (!EVP_PKEY_get_utf8_string_param(key, OSSL_PKEY_PARAM_GROUP_NAME,
                                     curve_name, sizeof(curve_name), &len)) {
    /* Error */
}
if (!EVP_PKEY_get_octet_string_param(key, OSSL_PKEY_PARAM_PUB_KEY,
                                     pub, sizeof(pub), &len)) {
    /* Error */
}
if (!EVP_PKEY_get_bn_param(key, OSSL_PKEY_PARAM_PRIV_KEY, &bn_priv)) {
    /* Error */
}

BN_clear_free(bn_priv);

```

SEE ALSO

[EVP_PKEY_CTX_new\(3\)](#), [provider-keymgmt\(7\)](#), [OSSL_PARAM\(3\)](#)

HISTORY

These functions were added in OpenSSL 3.0.

COPYRIGHT

Copyright 2020-2022 The OpenSSL Project Authors. All Rights Reserved.

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_GETTABLE_PARAMS(3ossl)