



*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 'OSSL\_DECODER\_from\_data.3oss1' command**

**\$ man OSSL\_DECODER\_from\_data.3oss1**

OSSL\_DECODER\_FROM\_BIO(3oss1)    OpenSSL    OSSL\_DECODER\_FROM\_BIO(3oss1)

### NAME

OSSL\_DECODER\_from\_data, OSSL\_DECODER\_from\_bio, OSSL\_DECODER\_from\_fp -  
Routines to perform a decoding

### SYNOPSIS

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

```
int OSSL_DECODER_from_bio(OSSL_DECODER_CTX *ctx, BIO *in);  
int OSSL_DECODER_from_fp(OSSL_DECODER_CTX *ctx, FILE *fp);  
int OSSL_DECODER_from_data(OSSL_DECODER_CTX *ctx, const unsigned char **pdata,  
                           size_t *pdata_len);
```

Feature availability macros:

OSSL\_DECODER\_from\_fp() is only available when OPENSSL\_NO\_STDIO is undefined.

### DESCRIPTION

OSSL\_DECODER\_from\_data() runs the decoding process for the context ctx, with input coming from \*pdata, \*pdata\_len bytes long. Both \*pdata and \*pdata\_len must be non-NULL. When OSSL\_DECODER\_from\_data() returns,

\*pdata is updated to point at the location after what has been decoded,  
and \*pdata\_len to have the number of remaining bytes.

OSSL\_DECODER\_from\_bio() runs the decoding process for the context ctx,  
with the input coming from the BIO in. Should it make a difference,  
it's recommended to have the BIO set in binary mode rather than text  
mode.

OSSL\_DECODER\_from\_fp() does the same thing as OSSL\_DECODER\_from\_bio(),  
except that the input is coming from the FILE fp.

## RETURN VALUES

OSSL\_DECODER\_from\_bio() and OSSL\_DECODER\_from\_fp() return 1 on success,  
or 0 on failure.

## EXAMPLES

To decode an RSA key encoded with PEM from a bio:

```
OSSL_DECODER_CTX *dctx;
EVP_PKEY *pkey = NULL;
const char *format = "PEM"; /* NULL for any format */
const char *structure = NULL; /* any structure */
const char *keytype = "RSA"; /* NULL for any key */
const unsigned char *pass = "my password";

dctx = OSSL_DECODER_CTX_new_for_pkey(&pkey, format, structure,
                                     keytype,
                                     OSSL_KEYMGMT_SELECT_KEYPAIR,
                                     NULL, NULL);

if (dctx == NULL) {
    /* error: no suitable potential decoders found */
}

if (pass != NULL)
```

```

    OSSL_DECODER_CTX_set_passphrase(dctx, pass, strlen(pass));
if (OSSL_DECODER_from_bio(dctx, bio)) {
    /* pkey is created with the decoded data from the bio */
} else {
    /* decoding failure */
}
OSSL_DECODER_CTX_free(dctx);

```

To decode an EC key encoded with DER from a buffer:

```

OSSL_DECODER_CTX *dctx;
EVP_PKEY *pkey = NULL;
const char *format = "DER"; /* NULL for any format */
const char *structure = NULL; /* any structure */
const char *keytype = "EC"; /* NULL for any key */
const unsigned char *pass = NULL
const unsigned char *data = buffer;
size_t datalen = sizeof(buffer);

dctx = OSSL_DECODER_CTX_new_for_pkey(&pkey, format, structure,
                                     keytype,
                                     OSSL_KEYMGMT_SELECT_KEYPAIR
                                     | OSSL_KEYMGMT_SELECT_DOMAIN_PARAMETERS,
                                     NULL, NULL);

if (dctx == NULL) {
    /* error: no suitable potential decoders found */
}

if (pass != NULL)
    OSSL_DECODER_CTX_set_passphrase(dctx, pass, strlen(pass));
if (OSSL_DECODER_from_data(dctx, &data, &datalen)) {
    /* pkey is created with the decoded data from the buffer */
} else {
    /* decoding failure */
}

```

```
}  
OSSL_DECODER_CTX_free(dctx);
```

## SEE ALSO

provider(7), OSSL\_DECODER\_CTX(3)

## HISTORY

The functions described here were added in OpenSSL 3.0.

## COPYRIGHT

Copyright 2020-2021 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    OSSL\_DECODER\_FROM\_BIO(3ossl)