



*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\_verify.3oss1' command**

**\$ man EVP\_PKEY\_verify.3oss1**

EVP\_PKEY\_VERIFY(3oss1)      OpenSSL      EVP\_PKEY\_VERIFY(3oss1)

### NAME

EVP\_PKEY\_verify\_init, EVP\_PKEY\_verify\_init\_ex, EVP\_PKEY\_verify -  
signature verification using a public key algorithm

### SYNOPSIS

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

```
int EVP_PKEY_verify_init(EVP_PKEY_CTX *ctx);
```

```
int EVP_PKEY_verify_init_ex(EVP_PKEY_CTX *ctx, const OSSL_PARAM params[]);
```

```
int EVP_PKEY_verify(EVP_PKEY_CTX *ctx,  
                  const unsigned char *sig, size_t siglen,  
                  const unsigned char *tbs, size_t tbslen);
```

### DESCRIPTION

EVP\_PKEY\_verify\_init() initializes a public key algorithm context ctx for signing using the algorithm given when the context was created using EVP\_PKEY\_CTX\_new(3) or variants thereof. The algorithm is used to fetch a EVP\_SIGNATURE method implicitly, see "Implicit fetch" in provider(7) for more information about implicit fetches.

EVP\_PKEY\_verify\_init\_ex() is the same as EVP\_PKEY\_verify\_init() but

additionally sets the passed parameters `params` on the context before returning.

The `EVP_PKEY_verify()` function performs a public key verification operation using `ctx`. The signature is specified using the `sig` and `siglen` parameters. The verified data (i.e. the data believed originally signed) is specified using the `tbs` and `tbslen` parameters.

## NOTES

After the call to `EVP_PKEY_verify_init()` algorithm specific control operations can be performed to set any appropriate parameters for the operation.

The function `EVP_PKEY_verify()` can be called more than once on the same context if several operations are performed using the same parameters.

## RETURN VALUES

`EVP_PKEY_verify_init()` and `EVP_PKEY_verify()` return 1 if the verification was successful and 0 if it failed. Unlike other functions the return value 0 from `EVP_PKEY_verify()` only indicates that the signature did not verify successfully (that is `tbs` did not match the original data or the signature was of invalid form) it is not an indication of a more serious error.

A negative value indicates an error other than signature verification failure. In particular a return value of -2 indicates the operation is not supported by the public key algorithm.

## EXAMPLES

Verify signature using PKCS#1 and SHA256 digest:

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

```

EVP_PKEY_CTX *ctx;
unsigned char *md, *sig;
size_t mdlen, siglen;
EVP_PKEY *verify_key;

/*
 * NB: assumes verify_key, sig, siglen md and mdlen are already set up
 * and that verify_key is an RSA public key
 */
ctx = EVP_PKEY_CTX_new(verify_key, NULL /* no engine */);
if (!ctx)
    /* Error occurred */
if (EVP_PKEY_verify_init(ctx) <= 0)
    /* Error */
if (EVP_PKEY_CTX_set_rsa_padding(ctx, RSA_PKCS1_PADDING) <= 0)
    /* Error */
if (EVP_PKEY_CTX_set_signature_md(ctx, EVP_sha256()) <= 0)
    /* Error */

/* Perform operation */
ret = EVP_PKEY_verify(ctx, sig, siglen, md, mdlen);

/*
 * ret == 1 indicates success, 0 verify failure and < 0 for some
 * other error.
 */

```

#### SEE ALSO

[EVP\\_PKEY\\_CTX\\_new\(3\)](#), [EVP\\_PKEY\\_encrypt\(3\)](#), [EVP\\_PKEY\\_decrypt\(3\)](#),  
[EVP\\_PKEY\\_sign\(3\)](#), [EVP\\_PKEY\\_verify\\_recover\(3\)](#), [EVP\\_PKEY\\_derive\(3\)](#)

The `EVP_PKEY_verify_init()` and `EVP_PKEY_verify()` functions were added in OpenSSL 1.0.0.

The `EVP_PKEY_verify_init_ex()` function was added in OpenSSL 3.0.

## COPYRIGHT

Copyright 2006-2021 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            EVP\_PKEY\_VERIFY(3ossl)