



*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 'openssl-ec.1ossl' command**

**\$ man openssl-ec.1ossl**

OPENSSL-EC(1ossl)            OpenSSL            OPENSSL-EC(1ossl)

### NAME

openssl-ec - EC key processing

### SYNOPSIS

```
openssl ec [-help] [-inform DER|PEM|P12|ENGINE] [-outform DER|PEM] [-in
filename|uri] [-passin arg] [-out filename] [-passout arg] [-des]
[-des3] [-idea] [-text] [-noout] [-param_out] [-pubin] [-pubout]
[-conv_form arg] [-param_enc arg] [-no_public] [-check] [-engine id]
[-provider name] [-provider-path path] [-propquery propq]
```

### DESCRIPTION

The openssl-ec(1) command processes EC keys. They can be converted between various forms and their components printed out. Note OpenSSL uses the private key format specified in 'SEC 1: Elliptic Curve Cryptography' (<http://www.secg.org/>). To convert an OpenSSL EC private key into the PKCS#8 private key format use the openssl-pkcs8(1) command.

### OPTIONS

-help

Print out a usage message.

-inform DER|PEM|P12|ENGINE

The key input format; unspecified by default. See openssl-format-options(1) for details.

-outform DER|PEM

The key output format; the default is PEM. See

`openssl-format-options(1)` for details.

Private keys are an SEC1 private key or PKCS#8 format. Public keys are a `SubjectPublicKeyInfo` as specified in IETF RFC 3280.

`-in filename|uri`

This specifies the input to read a key from or standard input if this option is not specified. If the key is encrypted a pass phrase will be prompted for.

`-out filename`

This specifies the output filename to write a key to or standard output by is not specified. If any encryption options are set then a pass phrase will be prompted for. The output filename should not be the same as the input filename.

`-passin arg, -passout arg`

The password source for the input and output file. For more information about the format of arg see `openssl-passphrase-options(1)`.

`-des|-des3|-idea`

These options encrypt the private key with the DES, triple DES, IDEA or any other cipher supported by OpenSSL before outputting it. A pass phrase is prompted for. If none of these options is specified the key is written in plain text. This means that using this command to read in an encrypted key with no encryption option can be used to remove the pass phrase from a key, or by setting the encryption options it can be use to add or change the pass phrase.

These options can only be used with PEM format output files.

`-text`

Prints out the public, private key components and parameters.

`-noout`

This option prevents output of the encoded version of the key.

`-param_out`

Print the elliptic curve parameters.

`-pubin`

By default, a private key is read from the input file. With this option a public key is read instead.

#### -pubout

By default a private key is output. With this option a public key will be output instead. This option is automatically set if the input is a public key.

#### -conv\_form arg

This specifies how the points on the elliptic curve are converted into octet strings. Possible values are: compressed, uncompressed (the default value) and hybrid. For more information regarding the point conversion forms please read the X9.62 standard. Note Due to patent issues the compressed option is disabled by default for binary curves and can be enabled by defining the preprocessor macro OPENSSL\_EC\_BIN\_PT\_COMP at compile time.

#### -param\_enc arg

This specifies how the elliptic curve parameters are encoded. Possible value are: named\_curve, i.e. the ec parameters are specified by an OID, or explicit where the ec parameters are explicitly given (see RFC 3279 for the definition of the EC parameters structures). The default value is named\_curve. Note the implicitlyCA alternative, as specified in RFC 3279, is currently not implemented in OpenSSL.

#### -no\_public

This option omits the public key components from the private key output.

#### -check

This option checks the consistency of an EC private or public key.

#### -engine id

See "Engine Options" in openssl(1). This option is deprecated.

#### -provider name

#### -provider-path path

#### -propquery propq

See "Provider Options" in openssl(1), provider(7), and property(7).

The `openssl-pkey(1)` command is capable of performing all the operations this command can, as well as supporting other public key types.

## EXAMPLES

The documentation for the `openssl-pkey(1)` command contains examples equivalent to the ones listed here.

To encrypt a private key using triple DES:

```
openssl ec -in key.pem -des3 -out keyout.pem
```

To convert a private key from PEM to DER format:

```
openssl ec -in key.pem -outform DER -out keyout.der
```

To print out the components of a private key to standard output:

```
openssl ec -in key.pem -text -noout
```

To just output the public part of a private key:

```
openssl ec -in key.pem -pubout -out pubkey.pem
```

To change the parameters encoding to explicit:

```
openssl ec -in key.pem -param_enc explicit -out keyout.pem
```

To change the point conversion form to compressed:

```
openssl ec -in key.pem -conv_form compressed -out keyout.pem
```

## SEE ALSO

`openssl(1)`, `openssl-pkey(1)`, `openssl-ecparam(1)`, `openssl-dsa(1)`,  
`openssl-rsa(1)`

## HISTORY

The `-engine` option was deprecated in OpenSSL 3.0.

The `-conv_form` and `-no_public` options are no longer supported with keys loaded from an engine in OpenSSL 3.0.

## COPYRIGHT

Copyright 2003-2022 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>>.