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-x509.1ossl' command

$ man openssl-x509.1ossl

OPENSSL-X509(1ossl) OpenSSL OPENSSL-X509(1ossl)

NAME

openssl-x509 - Certificate display and signing command

SYNOPSIS

openssl x509 [-help] [-in filename|uri] [-passin arg] [-new]
[-x509toreq] [-req] [-copy_extensions arg] [-inform DER|PEM] [-vfyopt
nm:v] [-key filename|uri] [-keyform DER|PEM|P12|ENGINE] [-signkey
filename|uri] [-out filename] [-outform DER|PEM] [-nocert] [-noout]
[-dateopt] [-text] [-certopt option] [-fingerprint] [-alias] [-serial]
[-startdate] [-enddate] [-dates] [-subject] [-issuer] [-nameopt option]
[-email] [-hash] [-subject_hash] [-subject_hash_old] [-issuer_hash]
[-issuer_hash_old] [-ext extensions] [-ocspid] [-ocsp_uri] [-purpose]
[-pubkey] [-modulus] [-checkend num] [-checkhost host] [-checkemail
host] [-checkip ipaddr] [-set_serial n] [-next_serial] [-days arg]
[-preserve_dates] [-subj arg] [-force_pubkey filename] [-clrext]
[-extfile filename] [-extensions section] [-sigopt nm:v] [-badsig]
[-digest] [-CA filename|uri] [-CAform DER|PEM|P12] [-CAkey
filename|uri] [-CAkeyform DER|PEM|P12|ENGINE] [-CAserial filename]
[-CAcreateserial] [-trustout] [-setalias arg] [-clrtrust] [-addtrust
arg] [-clrreject] [-addreject arg] [-rand files] [-writerand file]
[-engine id] [-provider name] [-provider-path path] [-propquery propq]

DESCRIPTION

This command is a multi-purposes certificate handling command. It can

be used to print certificate information, convert certificates to various forms, edit certificate trust settings, generate certificates from scratch or from certificating requests and then self-signing them or signing them like a "micro CA".

Since there are a large number of options they will split up into various sections.

OPTIONS

Input, Output, and General Purpose Options

-help

Print out a usage message.

-in filename|uri

This specifies the input to read a certificate from or the input file for reading a certificate request if the -req flag is used.

In both cases this defaults to standard input.

This option cannot be combined with the -new flag.

-passin arg

The key and certificate file password source. For more information about the format of arg see openssl-passphrase-options(1).

-new

Generate a certificate from scratch, not using an input certificate or certificate request. So the -in option must not be used in this case. Instead, the -subj option needs to be given. The public key to include can be given with the -force_pubkey option and defaults to the key given with the -key (or -signkey) option, which implies self-signature.

-x509toreq

Output a PKCS#10 certificate request (rather than a certificate).

The -key (or -signkey) option must be used to provide the private key for self-signing; the corresponding public key is placed in the subjectPKInfo field.

X.509 extensions included in a certificate input are not copied by default. X.509 extensions to be added can be specified using the -extfile option.

-req

By default a certificate is expected on input. With this option a PKCS#10 certificate request is expected instead, which must be correctly self-signed.

X.509 extensions included in the request are not copied by default. X.509 extensions to be added can be specified using the -extfile option.

-copy_extensions arg

Determines how to handle X.509 extensions when converting from a certificate to a request using the -x509toreq option or converting from a request to a certificate using the -req option. If arg is none or this option is not present then extensions are ignored. If arg is copy or copyall then all extensions are copied, except that subject identifier and authority key identifier extensions are not taken over when producing a certificate request.

The -ext option can be used to further restrict which extensions to copy.

-inform DER|PEM

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

-vfyopt nm:v

Pass options to the signature algorithm during verify operations. Names and values of these options are algorithm-specific.

-key filename|uri

This option provides the private key for signing a new certificate or certificate request. Unless -force_pubkey is given, the corresponding public key is placed in the new certificate or certificate request, resulting in a self-signature.

This option cannot be used in conjunction with the -CA option. It sets the issuer name to the subject name (i.e., makes it self-issued) and changes the public key to the supplied value (unless overridden by -force_pubkey). Unless the -preserve_dates option is supplied, it sets the validity start date to the current time and

the end date to a value determined by the -days option.

-signkey filename|uri

This option is an alias of -key.

-keyform DER|PEM|P12|ENGINE

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

-out filename

This specifies the output filename to write to or standard output by default.

-outform DER|PEM

The output format; the default is PEM. See openssl-format-options(1) for details.

-nocert

Do not output a certificate (except for printing as requested by below options).

-noout

This option prevents output except for printing as requested by below options.

Certificate Printing Options

Note: the -alias and -purpose options are also printing options but are described in the "Trust Settings" section.

-dateopt

Specify the date output format. Values are: rfc_822 and iso_8601. Defaults to rfc_822.

-text

Prints out the certificate in text form. Full details are printed including the public key, signature algorithms, issuer and subject names, serial number any extensions present and any trust settings.

-certopt option

Customise the print format used with -text. The option argument can be a single option or multiple options separated by commas. The -certopt switch may be also be used more than once to set multiple options. See the "Text Printing Flags" section for more

information.

-fingerprint

Calculates and prints the digest of the DER encoded version of the entire certificate (see digest options). This is commonly called a "fingerprint". Because of the nature of message digests, the fingerprint of a certificate is unique to that certificate and two certificates with the same fingerprint can be considered to be the same.

-alias

Prints the certificate "alias" (nickname), if any.

-serial

Prints the certificate serial number.

-startdate

Prints out the start date of the certificate, that is the notBefore date.

-enddate

Prints out the expiry date of the certificate, that is the notAfter date.

-dates

Prints out the start and expiry dates of a certificate.

-subject

Prints the subject name.

-issuer

Prints the issuer name.

-nameopt option

This specifies how the subject or issuer names are displayed. See openssl-namedisplay-options(1) for details.

-email

Prints the email address(es) if any.

-hash

Synonym for "-subject_hash" for backward compatibility reasons.

-subject_hash

Prints the "hash" of the certificate subject name. This is used in

OpenSSL to form an index to allow certificates in a directory to be looked up by subject name.

-subject_hash_old

Prints the "hash" of the certificate subject name using the older algorithm as used by OpenSSL before version 1.0.0.

-issuer_hash

Prints the "hash" of the certificate issuer name.

-issuer_hash_old

Prints the "hash" of the certificate issuer name using the older algorithm as used by OpenSSL before version 1.0.0.

-ext extensions

Prints out the certificate extensions in text form. Can also be used to restrict which extensions to copy. Extensions are specified with a comma separated string, e.g., "subjectAltName,subjectKeyIdentifier". See the x509v3_config(5) manual page for the extension names.

-ocspid

Prints the OCSP hash values for the subject name and public key.

-ocsp_uri

Prints the OCSP responder address(es) if any.

-purpose

This option performs tests on the certificate extensions and outputs the results. For a more complete description see "Certificate Extensions" in openssl-verification-options(1).

-pubkey

Prints the certificate's SubjectPublicKeyInfo block in PEM format.

-modulus

This option prints out the value of the modulus of the public key contained in the certificate.

Certificate Checking Options

-checkend arg

Checks if the certificate expires within the next arg seconds and exits nonzero if yes it will expire or zero if not.

-checkhost host

Check that the certificate matches the specified host.

-checkemail email

Check that the certificate matches the specified email address.

-checkip ipaddr

Check that the certificate matches the specified IP address.

Certificate Output Options

-set_serial n

Specifies the serial number to use. This option can be used with the -key, -signkey, or -CA options. If used in conjunction with the -CA option the serial number file (as specified by the -CAserial option) is not used.

The serial number can be decimal or hex (if preceded by "0x").

-next_serial

Set the serial to be one more than the number in the certificate.

-days arg

Specifies the number of days until a newly generated certificate expires. The default is 30. Cannot be used together with the -preserve_dates option.

-preserve_dates

When signing a certificate, preserve "notBefore" and "notAfter" dates of any input certificate instead of adjusting them to current time and duration. Cannot be used together with the -days option.

-subj arg

When a certificate is created set its subject name to the given value. When the certificate is self-signed the issuer name is set to the same value.

The arg must be formatted as "/type0=value0/type1=value1/type2=...". Special characters may be escaped by "\" (backslash), whitespace is retained. Empty values are permitted, but the corresponding type will not be included in the certificate. Giving a single "/" will lead to an empty sequence of RDNs (a NULL-DN). Multi-valued RDNs can be formed by

placing a "+" character instead of a "/" between the AttributeValueAssertions (AVAs) that specify the members of the set. Example: "/DC=org/DC=OpenSSL/DC=users/UID=123456+CN=John Doe"

This option can be used in conjunction with the -force_pubkey option to create a certificate even without providing an input certificate or certificate request.

-force_pubkey filename

When a certificate is created set its public key to the key in filename instead of the key contained in the input or given with the -key (or -signkey) option.

This option is useful for creating self-issued certificates that are not self-signed, for instance when the key cannot be used for signing, such as DH. It can also be used in conjunction with b<-new> and -subj to directly generate a certificate containing any desired public key.

-clrext

When transforming a certificate to a new certificate by default all certificate extensions are retained.

When transforming a certificate or certificate request, the -clrext option prevents taking over any extensions from the source. In any case, when producing a certificate request, neither subject identifier nor authority key identifier extensions are included.

-extfile filename

Configuration file containing certificate and request X.509 extensions to add.

-extensions section

The section in the extfile to add X.509 extensions from. If this option is not specified then the extensions should either be contained in the unnamed (default) section or the default section should contain a variable called "extensions" which contains the section to use. See the x509v3_config(5) manual page for details of the extension section format.

-sigopt nm:v

Pass options to the signature algorithm during sign operations.

This option may be given multiple times. Names and values provided using this option are algorithm-specific.

-badsig

Corrupt the signature before writing it; this can be useful for testing.

-digest

The digest to use. This affects any signing or printing option that uses a message digest, such as the -fingerprint, -key, and -CA options. Any digest supported by the openssl-dgst(1) command can be used. If not specified then SHA1 is used with -fingerprint or the default digest for the signing algorithm is used, typically SHA256.

Micro-CA Options

-CA filename|uri

Specifies the "CA" certificate to be used for signing. When present, this behaves like a "micro CA" as follows: The subject name of the "CA" certificate is placed as issuer name in the new certificate, which is then signed using the "CA" key given as detailed below.

This option cannot be used in conjunction with -key (or -signkey).

This option is normally combined with the -req option referencing a CSR. Without the -req option the input must be an existing certificate unless the -new option is given, which generates a certificate from scratch.

-CAform DER|PEM|P12,

The format for the CA certificate; unspecifed by default. See openssl-format-options(1) for details.

-CAkey filename|uri

Sets the CA private key to sign a certificate with. The private key must match the public key of the certificate given with -CA.

If this option is not provided then the key must be present in the

-CA input.

-CAkeyform DER|PEM|P12|ENGINE

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

-CAserial filename

Sets the CA serial number file to use.

When creating a certificate with this option and with the -CA option, the certificate serial number is stored in the given file. This file consists of one line containing an even number of hex digits with the serial number used last time. After reading this number, it is incremented and used, and the file is updated.

The default filename consists of the CA certificate file base name with .srl appended. For example if the CA certificate file is called mycacert.pem it expects to find a serial number file called mycacert.srl.

If the -CA option is specified and neither <-CAserial> or <-CAcreateserial> is given and the default serial number file does not exist, a random number is generated; this is the recommended practice.

-CAcreateserial

With this option and the -CA option the CA serial number file is created if it does not exist. A random number is generated, used for the certificate, and saved into the serial number file determined as described above.

Trust Settings

A trusted certificate is an ordinary certificate which has several additional pieces of information attached to it such as the permitted and prohibited uses of the certificate and possibly an "alias" (nickname).

Normally when a certificate is being verified at least one certificate must be "trusted". By default a trusted certificate must be stored locally and must be a root CA: any certificate chain ending in this CA is then usable for any purpose.

Trust settings currently are only used with a root CA. They allow a finer control over the purposes the root CA can be used for. For example, a CA may be trusted for SSL client but not SSL server use. See openssl-verification-options(1) for more information on the meaning of trust settings.

Future versions of OpenSSL will recognize trust settings on any certificate: not just root CAs.

-trustout

Mark any certificate PEM output as <trusted> certificate rather than ordinary. An ordinary or trusted certificate can be input but by default an ordinary certificate is output and any trust settings are discarded. With the -trustout option a trusted certificate is output. A trusted certificate is automatically output if any trust settings are modified.

-setalias arg

Sets the "alias" of the certificate. This will allow the certificate to be referred to using a nickname for example "Steve's Certificate".

-clrtrust

Clears all the permitted or trusted uses of the certificate.

-addtrust arg

Adds a trusted certificate use. Any object name can be used here but currently only clientAuth, serverAuth, emailProtection, and anyExtendedKeyUsage are defined. As of OpenSSL 1.1.0, the last of these blocks all purposes when rejected or enables all purposes when trusted. Other OpenSSL applications may define additional uses.

-clrreject

Clears all the prohibited or rejected uses of the certificate.

-addreject arg

Adds a prohibited trust anchor purpose. It accepts the same values as the -addtrust option.

-rand files, -writerand file

See "Random State Options" in openssl(1) for details.

-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).

Text Printing Flags

As well as customising the name printing format, it is also possible to customise the actual fields printed using the certopt option when the text option is present. The default behaviour is to print all fields.

compatible

Use the old format. This is equivalent to specifying no printing options at all.

no_header

Don't print header information: that is the lines saying "Certificate" and "Data".

no_version

Don't print out the version number.

no_serial

Don't print out the serial number.

no_signame

Don't print out the signature algorithm used.

no_validity

Don't print the validity, that is the notBefore and notAfter fields.

no_subject

Don't print out the subject name.

no_issuer

Don't print out the issuer name.

no_pubkey

Don't print out the public key.

no_sigdump

Don't give a hexadecimal dump of the certificate signature.

no_aux

Don't print out certificate trust information.

no_extensions

Don't print out any X509V3 extensions.

ext_default

Retain default extension behaviour: attempt to print out

unsupported certificate extensions.

ext_error

Print an error message for unsupported certificate extensions.

ext_parse

ASN1 parse unsupported extensions.

ext_dump

Hex dump unsupported extensions.

ca_default

The value used by openssl-ca(1), equivalent to no_issuer,

no_pubkey, no_header, and no_version.

EXAMPLES

Note: in these examples the '\' means the example should be all on one

line.

Print the contents of a certificate:

```
openssl x509 -in cert.pem -noout -text
```

Print the "Subject Alternative Name" extension of a certificate:

```
openssl x509 -in cert.pem -noout -ext subjectAltName
```

Print more extensions of a certificate:

```
openssl x509 -in cert.pem -noout -ext subjectAltName,nsCertType
```

Print the certificate serial number:

```
openssl x509 -in cert.pem -noout -serial
```

Print the certificate subject name:

```
openssl x509 -in cert.pem -noout -subject
```

Print the certificate subject name in RFC2253 form:

```
openssl x509 -in cert.pem -noout -subject -nameopt RFC2253
```

Print the certificate subject name in oneline form on a terminal supporting UTF8:

```
openssl x509 -in cert.pem -noout -subject -nameopt oneline,-esc_msb
```

Print the certificate SHA1 fingerprint:

```
openssl x509 -sha1 -in cert.pem -noout -fingerprint
```

Convert a certificate from PEM to DER format:

```
openssl x509 -in cert.pem -inform PEM -out cert.der -outform DER
```

Convert a certificate to a certificate request:

```
openssl x509 -x509toreq -in cert.pem -out req.pem -key key.pem
```

Convert a certificate request into a self-signed certificate using extensions for a CA:

```
openssl x509 -req -in careq.pem -extfile openssl.cnf -extensions v3_ca \
        -key key.pem -out cacert.pem
```

Sign a certificate request using the CA certificate above and add user certificate extensions:

```
openssl x509 -req -in req.pem -extfile openssl.cnf -extensions v3_usr \
        -CA cacert.pem -CAkey key.pem -CAcreateserial
```

Set a certificate to be trusted for SSL client use and change set its alias to "Steve's Class 1 CA"

```
openssl x509 -in cert.pem -addtrust clientAuth \
        -setalias "Steve's Class 1 CA" -out trust.pem
```

NOTES

The conversion to UTF8 format used with the name options assumes that T61Strings use the ISO8859-1 character set. This is wrong but Netscape and MSIE do this as do many certificates. So although this is incorrect it is more likely to print the majority of certificates correctly.

The -email option searches the subject name and the subject alternative name extension. Only unique email addresses will be printed out: it will not print the same address more than once.

BUGS

It is possible to produce invalid certificates or requests by specifying the wrong private key, using unsuitable X.509 extensions, or using inconsistent options in some cases: these should be checked.

There should be options to explicitly set such things as start and end dates rather than an offset from the current time.

SEE ALSO

openssl(1), openssl-req(1), openssl-ca(1), openssl-genrsa(1), openssl-gendsa(1), openssl-verify(1), x509v3_config(5)

HISTORY

The hash algorithm used in the -subject_hash and -issuer_hash options before OpenSSL 1.0.0 was based on the deprecated MD5 algorithm and the encoding of the distinguished name. In OpenSSL 1.0.0 and later it is based on a canonical version of the DN using SHA1. This means that any directories using the old form must have their links rebuilt using openssl-rehash(1) or similar.

The -signkey option has been renamed to -key in OpenSSL 3.0, keeping the old name as an alias.

The -engine option was deprecated in OpenSSL 3.0.

The -C option was removed in OpenSSL 3.0.

COPYRIGHT