



Rocky Enterprise Linux 9.2 Manual Pages on command 'CA.pl.1ssl'

C:\>man CA.pl.1ssl

CA.PL(1SSL) OpenSSL CA.PL(1SSL)

NAME

CA.pl - friendlier interface for OpenSSL certificate programs

SYNOPSIS

CA.pl -? | -h | -help

CA.pl -newcert | -newreq | -newreq-nodes | -xsign | -sign | -signCA | -signcert |

-crl | -newca [-extra-cmd extra-params]

CA.pl -pkcs12 [-extra-pkcs12 extra-params] [certname]

CA.pl -verify [-extra-verify extra-params] certfile...

CA.pl -revoke [-extra-ca extra-params] certfile [reason]

DESCRIPTION

The CA.pl script is a perl script that supplies the relevant command line arguments to the openssl command for some common certificate operations. It is intended to simplify the process of certificate creation and management by the use of some simple options.

OPTIONS

?, -h, -help

Prints a usage message.

-newcert

Creates a new self signed certificate. The private key is written to the file "newkey.pem" and the request written to the file "newreq.pem". This argument invokes openssl req command.

-newreq

Creates a new certificate request. The private key is written to the file "newkey.pem" and the request written to the file "newreq.pem". Executes openssl req command below the hood.

-newreq-nodes

Is like -newreq except that the private key will not be encrypted. Uses openssl req command.

-newca

Creates a new CA hierarchy for use with the ca program (or the -signcert and -xsign options). The user is prompted to enter the filename of the CA certificates (which should also contain the private key) or by hitting ENTER details of the CA will be prompted for. The relevant files and directories are created in a directory called "demoCA" in the current directory. openssl req and openssl ca commands are get invoked.

-pkcs12

Create a PKCS#12 file containing the user certificate, private key and CA certificate. It expects the user certificate and private key to be in the file "newcert.pem" and the CA certificate to be in the file demoCA/cacert.pem, it creates a file "newcert.p12". This command can thus be called after the -sign option. The PKCS#12 file can be imported directly into a browser. If there is an additional argument on the command line it will be used as the "friendly name" for the certificate (which is typically displayed in the browser list box), otherwise the name "My Certificate" is used. Delegates work to openssl pkcs12 command.

-sign, -signcert, -xsign

Calls the ca program to sign a certificate request. It expects the request to be in the file "newreq.pem". The new certificate is written to the file "newcert.pem" except in the case of the -xsign option when it is written to standard output. Leverages openssl ca command.

-signCA

This option is the same as the -signreq option except it uses the configuration file section v3_ca and so makes the signed request a valid CA certificate. This is useful when creating intermediate CA from a root CA. Extra params are

passed on to openssl ca command.

-signcert

This option is the same as -sign except it expects a self signed certificate to be present in the file "newreq.pem". Extra params are passed on to openssl x509 and openssl ca commands.

-crl

Generate a CRL. Executes openssl ca command.

-revoke certfile [reason]

Revoke the certificate contained in the specified certfile. An optional reason may be specified, and must be one of: unspecified, keyCompromise, CACompromise, affiliationChanged, superseded, cessationOfOperation, certificateHold, or removeFromCRL. Leverages openssl ca command.

-verify

Verifies certificates against the CA certificate for "demoCA". If no certificates are specified on the command line it tries to verify the file "newcert.pem". Invokes openssl verify command.

-extra-req | -extra-ca | -extra-pkcs12 | -extra-x509 | -extra-verify <extra-params>

The purpose of these parameters is to allow optional parameters to be supplied to openssl that this command executes. The -extra-cmd are specific to the option being used and the openssl command getting invoked. For example when this command invokes openssl req extra parameters can be passed on with the -extra-req parameter. The openssl commands being invoked per option are documented below. Users should consult openssl command documentation for more information.

EXAMPLES

Create a CA hierarchy:

```
CA.pl -newca
```

Complete certificate creation example: create a CA, create a request, sign the request and finally create a PKCS#12 file containing it.

```
CA.pl -newca
```

```
CA.pl -newreq
```

```
CA.pl -signreq
```

```
CA.pl -pkcs12 "My Test Certificate"
```

DSA CERTIFICATES

Although the CA.pl creates RSA CAs and requests it is still possible to use it with DSA certificates and requests using the req(1) command directly. The following example shows the steps that would typically be taken.

Create some DSA parameters:

```
openssl dsaparam -out dsap.pem 1024
```

Create a DSA CA certificate and private key:

```
openssl req -x509 -newkey dsa:dsap.pem -keyout cacert.pem -out cacert.pem
```

Create the CA directories and files:

```
CA.pl -newca
```

enter cacert.pem when prompted for the CA file name.

Create a DSA certificate request and private key (a different set of parameters can optionally be created first):

```
openssl req -out newreq.pem -newkey dsa:dsap.pem
```

Sign the request:

```
CA.pl -signreq
```

NOTES

Most of the filenames mentioned can be modified by editing the CA.pl script.

If the demoCA directory already exists then the -newca command will not overwrite it and will do nothing. This can happen if a previous call using the -newca option terminated abnormally. To get the correct behaviour delete the demoCA directory if it already exists.

Under some environments it may not be possible to run the CA.pl script directly (for example Win32) and the default configuration file location may be wrong. In this case the command:

```
perl -S CA.pl
```

can be used and the OPENSSL_CONF environment variable changed to point to the correct path of the configuration file.

The script is intended as a simple front end for the openssl program for use by a beginner. Its behaviour isn't always what is wanted. For more control over the behaviour of the certificate commands call the openssl command directly.

SEE ALSO

x509(1), ca(1), req(1), pkcs12(1), config(5)

COPYRIGHT

Copyright 2000-2017 The OpenSSL Project Authors. All Rights Reserved.

Licensed under the OpenSSL license (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>.

1.1.1f

2023-02-06

CA.PL(1SSL)