

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** *parameter*]

CA.pl **-pkcs12** [*certname*]

CA.pl **-verify** *certfile* ...

CA.pl **-revoke** *certfile* [*reason*]

DESCRIPTION

The **CA.pl** script is a perl script that supplies the relevant command line arguments to the **openssl**(1) 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.

The script is intended as a simple front end for the **openssl**(1) 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**(1) command directly.

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

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 can be set to point to the correct path of the configuration file.

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*. Invokes **openssl-req**(1).

-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**(1) under the hood.

-newreq-nodes

Is like **-newreq** except that the private key will not be encrypted. Uses **openssl-req**(1).

-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. Uses **openssl-r eq**(1) and **openssl-ca**(1).

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 directory if it already exists.

-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**(1).

-sign, -xsign

Calls the **openssl-ca**(1) command 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.

-signCA

This option is the same as the **-sign** 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 to **openssl-ca**(1).

-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 to **openssl-x509**(1) and **openssl-ca**(1).

-crl

Generate a CRL. Executes **openssl-ca**(1).

-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**(1).

-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**(1).

-extra-cmd parameter

For each option **extra-cmd**, pass *parameter* to the **openssl**(1) sub-command with the same name as *cmd*, if that sub-command is invoked. For example, if **openssl-req**(1) is invoked, the *parameter* given with **-extra-req** will be passed to it. For multi-word parameters, either repeat the option or quote the *parameters* so it looks like one word to your shell. See the individual 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 -sign
CA.pl -pkcs12 "My Test Certificate"
```

ENVIRONMENT

The environment variable **OPENSSL** may be used to specify the name of the OpenSSL program. It can be a full pathname, or a relative one.

The environment variable **OPENSSL_CONFIG** may be used to specify a configuration option and value to the **req** and **ca** commands invoked by this script. It's value should be the option and pathname, as in **-config /path/to/conf-file**.

SEE ALSO

openssl(1), **openssl-x509**(1), **openssl-ca**(1), **openssl-req**(1), **openssl-pkcs12**(1), **config**(5)

COPYRIGHT

Copyright 2000–2026 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>](https://www.openssl.org/source/license.html).