NAME

CA.pl - friendlier interface for OpenSSL certificate programs

SYNOPSIS

CA.pl [**-?**] [**-h**] [**-help**] [**-newcert**] [**-newreq**] [**-newreq-nodes**] [**-newca**] [**-xsign**] [**-sign**] [**-signreq**] [**-signcert**] [**-verify**] [**files**]

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.

COMMAND 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''.

-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''.

-newreq-nodes

is like **-newreq** except that the private key will not be encrypted.

-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.

-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.

-sign, **-signreq**, **-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.

-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.

-signcert

this option is the same as **-sign** except it expects a self signed certificate to be present in the file ''newreq.pem''.

-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''.

files

one or more optional certificate file names for use with the **-verify** command.

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 ‘‘openssl.cnf’’.

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.

ENVIRONMENT VARIABLES

The variable **OPENSSL_CONF** if defined allows an alternative configuration file location to be specified, it should contain the full path to the configuration file, not just its directory.

SEE ALSO

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