

NAME

openssl - OpenSSL command line tool

SYNOPSIS

openssl *command* [*command_opts*] [*command_args*]

openssl list [**standard-commands** | **digest-commands** | **cipher-commands** | **cipher-algorithms** | **digest-algorithms** | **public-key-algorithms**]

openssl no-XXX [*arbitrary options*]

DESCRIPTION

OpenSSL is a cryptography toolkit implementing the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) network protocols and related cryptography standards required by them.

The **openssl** program is a command line tool for using the various cryptography functions of OpenSSL's **crypto** library from the shell. It can be used for

- o Creation and management of private keys, public keys and parameters
- o Public key cryptographic operations
- o Creation of X.509 certificates, CSRs and CRLs
- o Calculation of Message Digests
- o Encryption and Decryption with Ciphers
- o SSL/TLS Client and Server Tests
- o Handling of S/MIME signed or encrypted mail
- o Time Stamp requests, generation and verification

COMMAND SUMMARY

The **openssl** program provides a rich variety of commands (*command* in the SYNOPSIS above), each of which often has a wealth of options and arguments (*command_opts* and *command_args* in the SYNOPSIS).

Many commands use an external configuration file for some or all of their arguments and have a **-config** option to specify that file. The environment variable **OPENSSL_CONF** can be used to specify the location of the file. If the environment variable is not specified, then the file is named **openssl.cnf** in the default certificate storage area, whose value depends on the configuration flags specified when the OpenSSL was built.

The list parameters **standard-commands**, **digest-commands**, and **cipher-commands** output a list (one entry per line) of the names of all standard commands, message digest commands, or cipher commands, respectively, that are available in the present **openssl** utility.

The list parameters **cipher-algorithms** and **digest-algorithms** list all cipher and message digest names, one entry per line. Aliases are listed as:

```
from => to
```

The list parameter **public-key-algorithms** lists all supported public key algorithms.

The command **no-XXX** tests whether a command of the specified name is available. If no command named *XXX* exists, it returns 0 (success) and prints **no-XXX**; otherwise it returns 1 and prints *XXX*. In both cases, the output goes to **stdout** and nothing is printed to **stderr**. Additional command line arguments are always ignored. Since for each cipher there is a command of the same name, this provides an easy way for shell scripts to test for the availability of ciphers in the **openssl** program. (**no-XXX** is not able to detect pseudo-commands such as **quit**, **list**, or **no-XXX** itself.)

Standard Commands**asn1parse**

Parse an ASN.1 sequence.

ca Certificate Authority (CA) Management.

ciphers

Cipher Suite Description Determination.

- cms**
CMS (Cryptographic Message Syntax) utility.
- crl** Certificate Revocation List (CRL) Management.
- crl2pkcs7**
CRL to PKCS#7 Conversion.
- dgst**
Message Digest Calculation.
- dh** Diffie-Hellman Parameter Management. Obsoleted by **dhparam**.
- dhparam**
Generation and Management of Diffie-Hellman Parameters. Superseded by **genpkey** and **pkeyparam**.
- dsa** DSA Data Management.
- dsaparam**
DSA Parameter Generation and Management. Superseded by **genpkey** and **pkeyparam**.
- ec** EC (Elliptic curve) key processing.
- ecparam**
EC parameter manipulation and generation.
- enc** Encoding with Ciphers.
- engine**
Engine (loadable module) information and manipulation.
- errstr**
Error Number to Error String Conversion.
- gendh**
Generation of Diffie-Hellman Parameters. Obsoleted by **dhparam**.
- gendsa**
Generation of DSA Private Key from Parameters. Superseded by **genpkey** and **pkey**.
- genpkey**
Generation of Private Key or Parameters.
- genrsa**
Generation of RSA Private Key. Superseded by **genpkey**.
- nseq**
Create or examine a Netscape certificate sequence.
- ocsp**
Online Certificate Status Protocol utility.
- passwd**
Generation of hashed passwords.
- pkcs12**
PKCS#12 Data Management.
- pkcs7**
PKCS#7 Data Management.
- pkcs8**
PKCS#8 format private key conversion tool.
- pkey**
Public and private key management.

pkeyparam

Public key algorithm parameter management.

pkeyutl

Public key algorithm cryptographic operation utility.

rand

Generate pseudo-random bytes.

rehash

Create symbolic links to certificate and CRL files named by the hash values.

req PKCS#10 X.509 Certificate Signing Request (CSR) Management.

rsa RSA key management.

rsautl

RSA utility for signing, verification, encryption, and decryption. Superseded by **pkeyutl**.

s_client

This implements a generic SSL/TLS client which can establish a transparent connection to a remote server speaking SSL/TLS. It's intended for testing purposes only and provides only rudimentary interface functionality but internally uses mostly all functionality of the OpenSSL **ssl** library.

s_server

This implements a generic SSL/TLS server which accepts connections from remote clients speaking SSL/TLS. It's intended for testing purposes only and provides only rudimentary interface functionality but internally uses mostly all functionality of the OpenSSL **ssl** library. It provides both an own command line oriented protocol for testing SSL functions and a simple HTTP response facility to emulate an SSL/TLS-aware webserver.

s_time

SSL Connection Timer.

sess_id

SSL Session Data Management.

smime

S/MIME mail processing.

speed

Algorithm Speed Measurement.

spkac

SPKAC printing and generating utility.

ts Time Stamping Authority tool (client/server).

verify

X.509 Certificate Verification.

version

OpenSSL Version Information.

x509

X.509 Certificate Data Management.

Message Digest Commands**md2**

MD2 Digest

md5

MD5 Digest

mdc2

MDC2 Digest

rmd160

RMD-160 Digest

sha SHA Digest**sha1**

SHA-1 Digest

sha224

SHA-224 Digest

sha256

SHA-256 Digest

sha384

SHA-384 Digest

sha512

SHA-512 Digest

Encoding and Cipher Commands**base64**

Base64 Encoding

bf bf-cbc bf-cfb bf-ecb bf-ofb

Blowfish Cipher

cast cast-cbc

CAST Cipher

cast5-cbc cast5-cfb cast5-ecb cast5-ofb

CAST5 Cipher

des des-cbc des-cfb des-ecb des-ede des-ede-cbc des-ede-cfb des-ede-ofb des-ofb

DES Cipher

des3 desx des-ede3 des-ede3-cbc des-ede3-cfb des-ede3-ofb

Triple-DES Cipher

idea idea-cbc idea-cfb idea-ecb idea-ofb

IDEA Cipher

rc2 rc2-cbc rc2-cfb rc2-ecb rc2-ofb

RC2 Cipher

rc4 RC4 Cipher**rc5 rc5-cbc rc5-cfb rc5-ecb rc5-ofb**

RC5 Cipher

OPTIONS

Details of which options are available depend on the specific command. This section describes some common options with common behavior.

Common Options**-help**

Provides a terse summary of all options.

Pass Phrase Options

Several commands accept password arguments, typically using **-passin** and **-passout** for input and output passwords respectively. These allow the password to be obtained from a variety of sources. Both of these options take a single argument whose format is described below. If no password argument is given and a password is required then the user is prompted to enter one: this will typically be read from the current

terminal with echoing turned off.

pass:password

The actual password is **password**. Since the password is visible to utilities (like 'ps' under Unix) this form should only be used where security is not important.

env:var

Obtain the password from the environment variable **var**. Since the environment of other processes is visible on certain platforms (e.g. ps under certain Unix OSes) this option should be used with caution.

file:pathname

The first line of **pathname** is the password. If the same **pathname** argument is supplied to **-passin** and **-passout** arguments then the first line will be used for the input password and the next line for the output password. **pathname** need not refer to a regular file: it could for example refer to a device or named pipe.

fd:number

Read the password from the file descriptor **number**. This can be used to send the data via a pipe for example.

stdin

Read the password from standard input.

SEE ALSO

asn1parse(1), ca(1), ciphers(1), cms(1), config(5), crl(1), crl2pkcs7(1), dgst(1), dhparam(1), dsa(1), dsaparam(1), ec(1), ecpkcs1(1), enc(1), engine(1), errstr(1), gendsa(1), genpkey(1), genrsa(1), nseq(1), ocsf(1), passwd(1), pkcs12(1), pkcs7(1), pkcs8(1), pkey(1), pkeyparam(1), pkeyutil(1), rand(1), rehash(1), req(1), rsa(1), rsautl(1), s_client(1), s_server(1), s_time(1), sess_id(1), smime(1), speed(1), spkac(1), ts(1), verify(1), version(1), x509(1), crypto(7), ssl(7), x509v3_config(5)

HISTORY

The **list-XXX-algorithms** pseudo-commands were added in OpenSSL 1.0.0; For notes on the availability of other commands, see their individual manual pages.

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>.