



*Full credit is given to the above companies including the OS that this PDF file was generated!*

### ***Rocky Enterprise Linux 9.2 Manual Pages on command 'X509\_load\_cert\_crl\_file.3ossl'***

***\$ man X509\_load\_cert\_crl\_file.3ossl***

X509\_LOOKUP\_HASH\_DIR(3ossl)    OpenSSL    X509\_LOOKUP\_HASH\_DIR(3ossl)

NAME

X509\_LOOKUP\_hash\_dir, X509\_LOOKUP\_file, X509\_LOOKUP\_store,  
X509\_load\_cert\_file\_ex, X509\_load\_cert\_file, X509\_load\_crl\_file,  
X509\_load\_cert\_crl\_file\_ex, X509\_load\_cert\_crl\_file - Default OpenSSL  
certificate lookup methods

SYNOPSIS

```
#include <openssl/x509_vfy.h>

X509_LOOKUP_METHOD *X509_LOOKUP_hash_dir(void);
X509_LOOKUP_METHOD *X509_LOOKUP_file(void);
X509_LOOKUP_METHOD *X509_LOOKUP_store(void);

int X509_load_cert_file_ex(X509_LOOKUP *ctx, const char *file, int type,
                           OSSL_LIB_CTX *libctx, const char *propq);
int X509_load_cert_file(X509_LOOKUP *ctx, const char *file, int type);
int X509_load_crl_file(X509_LOOKUP *ctx, const char *file, int type);
int X509_load_cert_crl_file_ex(X509_LOOKUP *ctx, const char *file, int type,
                               OSSL_LIB_CTX *libctx, const char *propq);
int X509_load_cert_crl_file(X509_LOOKUP *ctx, const char *file, int type);
```

## DESCRIPTION

X509\_LOOKUP\_hash\_dir and X509\_LOOKUP\_file are two certificate lookup methods to use with X509\_STORE, provided by OpenSSL library.

Users of the library typically do not need to create instances of these methods manually, they would be created automatically by

X509\_STORE\_load\_locations(3) or SSL\_CTX\_load\_verify\_locations(3) functions.

Internally loading of certificates and CRLs is implemented via functions X509\_load\_cert\_crl\_file, X509\_load\_cert\_file and X509\_load\_crl\_file. These functions support parameter type, which can be one of constants FILETYPE\_PEM, FILETYPE\_ASN1 and FILETYPE\_DEFAULT.

They load certificates and/or CRLs from specified file into memory cache of X509\_STORE objects which given ctx parameter is associated with.

Functions X509\_load\_cert\_file and X509\_load\_crl\_file can load both PEM and DER formats depending of type value. Because DER format cannot contain more than one certificate or CRL object (while PEM can contain several concatenated PEM objects) X509\_load\_cert\_crl\_file with FILETYPE\_ASN1 is equivalent to X509\_load\_cert\_file.

Constant FILETYPE\_DEFAULT with NULL filename causes these functions to load default certificate store file (see X509\_STORE\_set\_default\_paths(3)).

Functions return number of objects loaded from file or 0 in case of error.

Both methods support adding several certificate locations into one X509\_STORE.

This page documents certificate store formats used by these methods and caching policy.

### File Method

The X509\_LOOKUP\_file method loads all the certificates or CRLs present in a file into memory at the time the file is added as a lookup source.

File format is ASCII text which contains concatenated PEM certificates and CRLs.

This method should be used by applications which work with a small set of CAs.

#### Hashed Directory Method

X509\_LOOKUP\_hash\_dir is a more advanced method, which loads certificates and CRLs on demand, and caches them in memory once they are loaded. As of OpenSSL 1.0.0, it also checks for newer CRLs upon each lookup, so that newer CRLs are as soon as they appear in the directory.

The directory should contain one certificate or CRL per file in PEM format, with a filename of the form hash.N for a certificate, or hash.rN for a CRL. The hash is the value returned by the X509\_NAME\_hash\_ex(3) function applied to the subject name for certificates or issuer name for CRLs. The hash can also be obtained via the -hash option of the openssl-x509(1) or openssl-crl(1) commands. The .N or .rN suffix is a sequence number that starts at zero, and is incremented consecutively for each certificate or CRL with the same hash value. Gaps in the sequence numbers are not supported, it is assumed that there are no more objects with the same hash beyond the first missing number in the sequence.

Sequence numbers make it possible for the directory to contain multiple certificates with same subject name hash value. For example, it is possible to have in the store several certificates with same subject or several CRLs with same issuer (and, for example, different validity period).

When checking for new CRLs once one CRL for given hash value is loaded, hash\_dir lookup method checks only for certificates with sequence number greater than that of the already cached CRL.

Note that the hash algorithm used for subject name hashing changed in OpenSSL 1.0.0, and all certificate stores have to be rehashed when moving from OpenSSL 0.9.8 to 1.0.0.

OpenSSL includes a openssl-rehash(1) utility which creates symlinks with hashed names for all files with .pem suffix in a given directory.

#### OSL\_STORE Method

X509\_LOOKUP\_store is a method that allows access to any store of certificates and CRLs through any loader supported by `ossl_store(7)`.

It works with the help of URIs, which can be direct references to certificates or CRLs, but can also be references to catalogues of such objects (that behave like directories).

This method overlaps the "File Method" and "Hashed Directory Method" because of the 'file:' scheme loader. It does no caching of its own, but can use a caching `ossl_store(7)` loader, and therefore depends on the loader's capability.

## RETURN VALUES

`X509_LOOKUP_hash_dir()`, `X509_LOOKUP_file()` and `X509_LOOKUP_store()` always return a valid `X509_LOOKUP_METHOD` structure.

`X509_load_cert_file()`, `X509_load_crl_file()` and

`X509_load_cert_crl_file()` return the number of loaded objects or 0 on error.

## SEE ALSO

`PEM_read_PrivateKey(3)`, `X509_STORE_load_locations(3)`,  
`SSL_CTX_load_verify_locations(3)`, `X509_LOOKUP_meth_new(3)`,  
`ossl_store(7)`

## HISTORY

The functions `X509_load_cert_file_ex()`, `X509_load_cert_crl_file_ex()` and `X509_LOOKUP_store()` were added in OpenSSL 3.0.

## COPYRIGHT

Copyright 2015-2021 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>>.

3.0.7                    2023-07-13    X509\_LOOKUP\_HASH\_DIR(3ossl)