

NAME

CURLOPT_SSLCERT – SSL client certificate

SYNOPSIS

```
#include <curl/curl.h>
```

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_SSLCERT, char *cert);
```

DESCRIPTION

Pass a pointer to a null-terminated string as parameter. The string should be the filename of your client certificate. The default format is *PEM* but can be changed with *CURLOPT_SSLCERTTYPE(3)*.

(Schannel) Client certificates can be specified by a path expression to a certificate store. (You can import *PFX* to a store first). You can use "<store location>\<store name>\<thumbprint>" to refer to a certificate in the system certificates store, for example, "**CurrentUser\MY\934a7ac6f8a5d5**". The thumbprint is usually a SHA-1 hex string which you can see in certificate details. Following store locations are supported: **CurrentUser**, **LocalMachine**, **CurrentService**, **Services**, **CurrentUserGroupPolicy**, **LocalMachineGroupPolicy**, **LocalMachineEnterprise**. Schannel also support P12 certificate file, with the string *P12* specified with *CURLOPT_SSLCERTTYPE(3)*.

When using a client certificate, you most likely also need to provide a private key with *CURLOPT_SSLKEY(3)*.

The application does not have to keep the string around after setting this option.

Using this option multiple times makes the last set string override the previous ones. Set it to *NULL* to disable its use again.

DEFAULT

NULL

PROTOCOLS

This functionality affects all TLS based protocols: HTTPS, FTPS, IMAPS, POP3S, SMTPS etc.

All TLS backends support this option.

EXAMPLE

```
int main(void)
{
    CURL *curl = curl_easy_init();
    if(curl) {
        CURLcode result;
        curl_easy_setopt(curl, CURLOPT_URL, "https://example.com/");
        curl_easy_setopt(curl, CURLOPT_SSLCERT, "client.pem");
        curl_easy_setopt(curl, CURLOPT_SSLKEY, "key.pem");
        curl_easy_setopt(curl, CURLOPT_KEYPASSWD, "s3cret");
        result = curl_easy_perform(curl);
        curl_easy_cleanup(curl);
    }
}
```

AVAILABILITY

Added in curl 7.1

RETURN VALUE

curl_easy_setopt(3) returns a *CURLcode* indicating success or error.

CURLE_OK (0) means everything was OK, non-zero means an error occurred, see *libcurl-errors(3)*.

SEE ALSO**CURLOPT_KEYPASSWD(3), CURLOPT_SSLCERTTYPE(3), CURLOPT_SSLKEY(3)**