NAME

CURLOPT_SSLKEY − private key file for TLS and SSL client cert

SYNOPSIS

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

CURLcode curl_easy_setopt(CURL *handle, CURLOPT_SSLKEY, char *keyfile);
```

DESCRIPTION

Pass a pointer to a null−terminated string as parameter. The string should be the filename of your private key. The default format is "PEM" and can be changed with *CURLOPT_SSLKEYTYPE(3)*.

(Windows) This option is ignored by the Schannel SSL backend because it expects the private key to be already present in the key−chain or PKCS#12 file containing the certificate.

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.

This option works only with the following TLS backends: OpenSSL, Rustls, Schannel, mbedTLS and wolfSSL

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 7.9.3

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_SSLCERT(3), **CURLOPT_SSLKEYTYPE**(3), **CURLOPT_SSLKEY_BLOB**(3)