NAME

CURLOPT_SSLCERT_BLOB − SSL client certificate from memory blob

SYNOPSIS

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

CURLcode curl_easy_setopt(CURL *handle, CURLOPT_SSLCERT_BLOB,
                          struct curl_blob *stblob);
```

DESCRIPTION

Pass a pointer to a curl_blob structure, which contains (pointer and size) a client certificate. The format must be "P12" on Schannel. The format must be "P12" or "PEM" on OpenSSL. The format must be "DER" or "PEM" on mbedTLS. The format must be specified with *CURLOPT_SSLCERTTYPE(3)*.

If the blob is initialized with the flags member of struct curl_blob set to CURL_BLOB_COPY, the application does not have to keep the buffer around after setting this.

This option is an alternative to *CURLOPT_SSLCERT(3)* which instead expects a filename as input.

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, Schannel, mbedTLS and wolfSSL

EXAMPLE

```
extern char *certificateData; /* point to data */
extern size_t filesize; /* size of data */

int main(void)
{
 CURL *curl = curl_easy_init();
 if(curl) {
  CURLcode result;
  struct curl_blob stblob;
  stblob.data = certificateData;
  stblob.len = filesize;
  stblob.flags = CURL_BLOB_COPY;
  curl_easy_setopt(curl, CURLOPT_URL, "https://example.com/");
  curl_easy_setopt(curl, CURLOPT_SSLCERT_BLOB, &stblob);
  curl_easy_setopt(curl, CURLOPT_SSLCERTTYPE, "P12");
  curl_easy_setopt(curl, CURLOPT_KEYPASSWD, "s3cret");
  result = curl_easy_perform(curl);
  curl_easy_cleanup(curl);
 }
}
```

AVAILABILITY

Added in curl 7.71.0

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)