NAME

CURLOPT_PROXY_CAINFO_BLOB − proxy Certificate Authority (CA) bundle in PEM format

SYNOPSIS

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

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

DESCRIPTION

This option is for connecting to an HTTPS proxy, not an HTTPS server.

Pass a pointer to a curl_blob structure, which contains information (pointer and size) about a memory block with binary data of PEM encoded content holding one or more certificates to verify the HTTPS proxy with.

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.

If *CURLOPT_PROXY_SSL_VERIFYPEER(3)* is zero and you avoid verifying the server's certificate, *CURLOPT_PROXY_CAINFO_BLOB(3)* is not needed.

This option overrides *CURLOPT_PROXY_CAINFO(3)*.

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 and Schannel

EXAMPLE

```
#include <string.h> /* for strlen */

extern char *strpem; /* strpem must point to a PEM string */
int main(void)
{
  CURL *curl = curl_easy_init();
  if(curl) {
    CURLcode result;
    struct curl_blob blob;
    curl_easy_setopt(curl, CURLOPT_URL, "https://example.com/");
    /* using an HTTPS proxy */
    curl_easy_setopt(curl, CURLOPT_PROXY, "https://proxy.example:443");
    blob.data = strpem;
    blob.len = strlen(strpem);
    blob.flags = CURL_BLOB_COPY;
    curl_easy_setopt(curl, CURLOPT_PROXY_CAINFO_BLOB, &blob);
    result = curl_easy_perform(curl);
    curl_easy_cleanup(curl);
  }
}
```

AVAILABILITY

Added in curl 7.77.0

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_CAINFO(3), **CURLOPT_CAINFO_BLOB**(3), **CURLOPT_CAPATH**(3), **CURLOPT_PROXY_CAINFO**(3), **CURLOPT_PROXY_CAPATH**(3), **CURLOPT_PROXY_SSL_VERIFYHOST**(3), **CURLOPT_PROXY_SSL_VERIFYPEER**(3), **CURLOPT_SSL_VERIFYHOST**(3), **CURLOPT_SSL_VERIFYPEER**(3)