NAME

CURLOPT_PROXY_ISSUERCERT_BLOB − proxy issuer SSL certificate from memory blob

SYNOPSIS

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

CURLcode curl_easy_setopt(CURL *handle, CURLOPT_PROXY_ISSUERCERT_BLOB,
                          struct curl_blob *blob);
```

DESCRIPTION

Pass a pointer to a curl_blob struct, which contains information (pointer and size) about a memory block with binary data of a CA certificate in PEM format. If the option is set, an additional check against the peer certificate is performed to verify the issuer of the HTTPS proxy is indeed the one associated with the certificate provided by the option. This additional check is useful in multi−level PKI where one needs to enforce that the peer certificate is from a specific branch of the tree.

This option should be used in combination with the *CURLOPT_PROXY_SSL_VERIFYPEER(3)* option. Otherwise, the result of the check is not considered as failure.

A specific error code (CURLE_SSL_ISSUER_ERROR) is defined with the option, which is returned if the setup of the SSL/TLS session has failed due to a mismatch with the issuer of peer certificate (*CURLOPT_PROXY_SSL_VERIFYPEER(3)* has to be set too for the check to fail).

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_PROXY_ISSUERCERT(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

EXAMPLE

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

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 = certificateData;
  blob.len = filesize;
  blob.flags = CURL_BLOB_COPY;
  curl_easy_setopt(curl, CURLOPT_PROXY_ISSUERCERT_BLOB, &blob);
  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_ISSUERCERT_BLOB(3), **CURLOPT_PROXY_SSL_VERIFYHOST**(3), **CURLOPT_PROXY_SSL_VERIFYPEER**(3), **CURLOPT_SSL_VERIFYHOST**(3), **CURLOPT_SSL_VERIFYPEER**(3)