

NAME

CURLOPT_PROXY_SSL_VERIFYHOST – verify the proxy certificate's name against host

SYNOPSIS

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

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_PROXY_SSL_VERIFYHOST,
                           long verify);
```

DESCRIPTION

Pass a long set to 2L as asking curl to *verify* in the HTTPS proxy's certificate name fields against the proxy name.

This option determines whether libcurl verifies that the proxy cert contains the correct name for the name it is known as.

When *CURLOPT_PROXY_SSL_VERIFYHOST(3)* is 2, the proxy certificate must indicate that the server is the proxy to which you meant to connect to, or the connection fails.

curl considers the proxy the intended one when the Common Name field or a Subject Alternate Name field in the certificate matches the hostname in the proxy string which you told curl to use.

If *verify* value is set to 1:

In 7.28.0 and earlier: treated as a debug option of some sorts, not supported anymore due to frequently leading to programmer mistakes.

From 7.28.1 to 7.65.3: setting it to 1 made *curl_easy_setopt(3)* return an error and leaving the flag untouched.

From 7.66.0: treats 1 and 2 the same.

When the *verify* value is 0L, the connection succeeds regardless of the names used in the certificate. Use that ability with caution.

See also *CURLOPT_PROXY_SSL_VERIFYPEER(3)* to verify the digital signature of the proxy certificate.

DEFAULT

2

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");

        /* Set the default value: strict name check please */
        curl_easy_setopt(curl, CURLOPT_PROXY_SSL_VERIFYHOST, 2L);

        result = curl_easy_perform(curl);
```

```
    curl_easy_cleanup(curl);  
  }  
}
```

AVAILABILITY

Added in curl 7.52.0

RETURN VALUE

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

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

SEE ALSO

CURL_...CAINFO(3), CURL_...PROXY_CAINFO(3), CURL_...PROXY_SSL_VERIFYPEER(3), CURL_...SSL_VERIFYPEER(3)