

NAME

CURLOPT_PROXY_SSLVERSION – preferred HTTPS proxy TLS version

SYNOPSIS

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

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_PROXY_SSLVERSION,  
                           long version);
```

DESCRIPTION

Pass a long as parameter to control which version of SSL/TLS to attempt to use when connecting to an HTTPS proxy.

Use one of the available defines for this purpose. The available options are:

CURL_SSLVERSION_DEFAULT

The default action. This attempts to figure out the remote SSL protocol version.

CURL_SSLVERSION_TLSv1

TLSv1.x

CURL_SSLVERSION_TLSv1_0

TLSv1.0

CURL_SSLVERSION_TLSv1_1

TLSv1.1

CURL_SSLVERSION_TLSv1_2

TLSv1.2

CURL_SSLVERSION_TLSv1_3

TLSv1.3

The maximum TLS version can be set by using *one* of the CURL_SSLVERSION_MAX_ macros below. It is also possible to OR *one* of the CURL_SSLVERSION_ macros with *one* of the CURL_SSLVERSION_MAX_ macros. The MAX macros are not supported for wolfSSL.

CURL_SSLVERSION_MAX_DEFAULT

The flag defines the maximum supported TLS version as TLSv1.2, or the default value from the SSL library.

CURL_SSLVERSION_MAX_TLSv1_0

The flag defines maximum supported TLS version as TLSv1.0.

CURL_SSLVERSION_MAX_TLSv1_1

The flag defines maximum supported TLS version as TLSv1.1.

CURL_SSLVERSION_MAX_TLSv1_2

The flag defines maximum supported TLS version as TLSv1.2.

CURL_SSLVERSION_MAX_TLSv1_3

The flag defines maximum supported TLS version as TLSv1.3.

DEFAULT

CURL_SSLVERSION_DEFAULT

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");

    /* ask libcurl to use TLS version 1.0 or later */
    curl_easy_setopt(curl, CURLOPT_SSLVERSION, CURL_SSLVERSION_TLSv1);

    /* Perform the request */
    result = curl_easy_perform(curl);
    curl_easy_cleanup(curl);
}
```

HISTORY

CURL_SSLVERSION_* macros became *long* types in 8.16.0, prior to this version a *long* cast was necessary when passed to *curl_easy_setopt(3)*.

AVAILABILITY

Added in curl 7.52.0

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_HTTP_VERSION(3), **CURLOPT_IPRESOLVE(3)**, **CURLOPT_SSLVERSION(3)**,
CURLOPT_USE_SSL(3)