

NAME

CURLOPT_FTP_SSL_CCC – switch off SSL again with FTP after auth

SYNOPSIS

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

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_FTP_SSL_CCC,
                           long how);
```

DESCRIPTION

If enabled, this option makes libcurl use CCC (Clear Command Channel). It shuts down the SSL/TLS layer after authenticating. The rest of the control channel communication remains unencrypted. This allows NAT routers to follow the FTP transaction. Pass a long using one of the values below

CURLFTPSSL_CCC_NONE
do not attempt to use CCC.

CURLFTPSSL_CCC_PASSIVE
Do not initiate the shutdown, but wait for the server to do it. Do not send a reply.

CURLFTPSSL_CCC_ACTIVE
Initiate the shutdown and wait for a reply.

DEFAULT

CURLFTPSSL_CCC_NONE

PROTOCOLS

This functionality affects ftp only

EXAMPLE

```
int main(void)
{
    CURL *curl = curl_easy_init();
    if(curl) {
        CURLcode result;
        curl_easy_setopt(curl, CURLOPT_URL, "ftp://example.com/file.txt");
        curl_easy_setopt(curl, CURLOPT_USE_SSL, CURLUSESSL_CONTROL);
        /* go back to clear-text FTP after authenticating */
        curl_easy_setopt(curl, CURLOPT_FTP_SSL_CCC, CURLFTPSSL_CCC_ACTIVE);
        result = curl_easy_perform(curl);
        curl_easy_cleanup(curl);
    }
}
```

HISTORY

CURLFTPSSL_* enums 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 7.16.1

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_FTPSSLAUTH(3), **CURLOPT_PROTOCOLS_STR(3)**, **CURLOPT_USE_SSL(3)**