

NAME

CURLOPT_USERPWD – username and password to use in authentication

SYNOPSIS

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

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_USERPWD, char *userpwd);
```

DESCRIPTION

Pass a char pointer as parameter, pointing to a null-terminated login details string for the connection. The format of which is: [username]:[password].

When using Kerberos V5 authentication with a Windows based server, you should specify the username part with the domain name in order for the server to successfully obtain a Kerberos Ticket. If you do not then the initial part of the authentication handshake may fail.

When using NTLM, the username can be specified without the domain name should the server be part of a single domain and forest.

To specify the domain name use either Down-Level Logon Name or UPN (User Principal Name) formats. For example **EXAMPLE** ~~ser~~ and **user@example.com** respectively.

Some HTTP servers (on Windows) support inclusion of the domain for Basic authentication as well.

When using HTTP and *CURLOPT_FOLLOWLOCATION(3)*, libcurl might perform several requests to possibly different hosts. libcurl only sends this user and password information to hosts using the initial host-name (unless *CURLOPT_UNRESTRICTED_AUTH(3)* is set), so if libcurl follows redirects to other hosts, it does not send the user and password to those. This is enforced to prevent accidental information leakage.

Use *CURLOPT_HTTPAUTH(3)* to specify the authentication method for HTTP based connections or *CURLOPT_LOGIN_OPTIONS(3)* to control IMAP, POP3 and SMTP options.

The user and password strings are not URL decoded, so there is no way to send in a username containing a colon using this option. Use *CURLOPT_USERNAME(3)* for that, or include it in the URL.

The application does not have to keep the string around after setting this option.

Using this option multiple times makes the last set string override the previous ones. Set it to NULL to disable its use again.

DEFAULT

NULL

PROTOCOLS

This functionality affects all supported protocols

EXAMPLE

```
int main(void)
{
    CURL *curl = curl_easy_init();
    if(curl) {
        CURLcode result;
        curl_easy_setopt(curl, CURLOPT_URL, "https://example.com/foo.bin");

        curl_easy_setopt(curl, CURLOPT_USERPWD, "clark:kent");

        result = curl_easy_perform(curl);
```

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

AVAILABILITY

Added in curl 7.1

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_PASSWORD(3), **CURLOPT_PROXYUSERPWD(3)**, **CURLOPT_USERNAME(3)**