

NAME

CURLOPT_USERPWD – set user + password to work with

SYNOPSIS

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

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

DESCRIPTION

Pass a char * as parameter, pointing to a zero terminated login details string for the connection. The format of which is: [user name]:[password].

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

When using NTLM, the user name can be specified simply as the user name 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\user 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 will only send this user and password information to hosts using the initial host name (unless *CURLOPT_UNRESTRICTED_AUTH(3)* is set), so if libcurl follows locations to other hosts it will 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's no way to send in a user name containing a colon using this option. Use *CURLOPT_USERNAME(3)* for that, or include it in the URL.

DEFAULT

NULL

PROTOCOLS

Most

EXAMPLE

TODO

AVAILABILITY

Always

RETURN VALUE

Returns CURLE_OK on success or CURLE_OUT_OF_MEMORY if there was insufficient heap space.

SEE ALSO

CURLOPT_USERNAME(3), CURLOPT_PASSWORD(3),